**Formal Methods in System Design manuscript No.**
(will be inserted by the editor)

# Symbolic Unfolding of Parametric Stopwatch Petri Nets

**Claude Jard · Didier Lime · Olivier H. Roux · Louis-Marie Traonouez**

Received: date / Accepted: date

**Abstract** We address the problem of unfolding safe parametric stopwatch time Petri nets (PSwPNs), i.e., safe time Petri nets (TPNs) possibly extended with time parameters and stopwatches. We extend the notion of branching process to account for the dates of the occurrences of events and thus define a symbolic unfolding for PSwPNs. In the case of TPNs we also propose a method based on our so-called time branching processes to compute a finite complete prefix of the symbolic unfolding. The originality of our work relies on a precise handling of direct conflicts between events, and the analysis of their effects on the constraints between the firing dates of those events.

**Keywords** Time Petri nets · Unfoldings · Stopwatches · Parameters

## 1 Introduction

The analysis of concurrent systems is one of the most challenging practical problems in computer science. Formal specification using Petri nets has the advantage to focus on the tricky part of such systems, that is parallelism, synchronization, conflicts and timing aspects. Among the different analysis techniques, we chose to develop the work on unfoldings [13].

This work has been partially funded by ANR project ImpRo (ANR-2010-BLAN-0317)

Claude Jard
Université de Nantes, LINA CNRS UMR 6241, Nantes, France
E-mail: Claude.Jard@univ-nantes.fr

Didier Lime and Olivier H. Roux
École Centrale de Nantes, IRCCyN CNRS UMR 6597, Nantes, France
E-mail: {Didier.Lime,Olivier-h.Roux}@irccyn.ec-nantes.fr

Louis-Marie Traonouez
INRIA, IRISA CNRS UMR 6074, Rennes, France
E-mail: Louis-Marie.Traonouez@inria.fr

Unfoldings were introduced in the early 1990s as a mathematical model of causality and became popular in the domain of computer aided verification. The main reason was to speed up the standard model-checking technique based on the computation of the interleavings of actions, leading to a very large state space in case of highly concurrent systems. The seminal papers are [25] and [12]. They dealt with basic bounded Petri nets.

Since then, the technique has attracted more attention, and the notion of unfolding has been extended to more expressive classes of Petri nets (Petri nets with read and inhibitor arcs [29,3], unbounded nets [1], high-level nets [19], and time Petri nets [9]). It has also been applied to networks of timed automata [8,7].

Advancing this line of works, we present in this paper a method to unfold safe parametric stopwatch Petri nets. Stopwatch Petri nets (SwPNs) [5] are a strict extension of the classical time Petri nets *à la* Merlin (TPNs) [26,4] and provide a means to model the suspension and resumption of actions with a memory of the "work" done before the suspension. This is very useful to model real-time preemptive scheduling policies for example [20,22].

The contribution of this paper is a new unfolding algorithm addressing the problem for stopwatch and parametric models for the first time. When applied to the subclass of time Petri nets, it provides an alternative to [9] and improves on the latter method by providing a more compact unfolding and not requiring read arcs in the unfolding (if the TPN itself has no read arcs of course). We also provide a way to compute a finite complete prefix of the unfolding for (safe) TPNs. Note this is the best we can do as most interesting properties, such as reachability, are undecidable in time Petri nets in presence of stopwatches [5] or parameters [28]. A preliminary version of this paper was published as [27].

While not extremely difficult from a theoretical point of view, we think that the handling of parameters is of utmost practical importance: adding parameters in specifications is a real need. It is often difficult to set them a priori: indeed, we expect from the analysis some useful information about their possible values. This feature of genericity clearly adds some "robustness" to the modeling phase. It is important to note that, as for time, we handle these parameters symbolically to achieve this genericity and the unfolding technique synthesizes all their possible values as linear constraint expressions.

Finally, note that the lack of existence of a finite prefix in the stopwatch or parametric cases is not necessarily prohibitive as several analysis techniques, such as supervision, can do without it [17]. Practical experience also demonstrates that even for very expressive models, such as Linear Hybrid Automata [18], the undecidability of the interesting problems still allows to analyze them in many cases.

This article is organized as follows: Section 2 recalls the basics of Petri nets with read arcs, branching processes, and unfoldings. Section 3 presents the time Petri net with read arcs model, and its concurrent semantics as time processes. Section 3 introduces our symbolic unfolding approach in the simpler case of time Petri nets, through the definition of time branching processes.

Section 5 defines a finite complete prefix of the symbolic unfolding for TPNs, based on time branching processes. Finally, Section 6 extends the notions of time processes, time branching processes and symbolic unfoldings to take time parameters and stopwatches into account.

## 2 Petri Nets and Branching Processes

### 2.1 Petri Nets

**Definition 1 (Petri Net)** A (safe) *Petri Net* with read arcs is a 5-tuple $(P, T, F, F_r, m_0)$ where:

- $P$ is a finite set of *places*,
- $T$ is a finite set of *transitions*, with $P \cap T = \emptyset$,
- $F \subseteq (P \times T) \cup (T \times P)$ is the *transition relation*,
- $F_r \subseteq P \times T$ is the *read relation*,
- $m_0 \subseteq P$ is called *initial marking*.

This structure defines an alternated bipartite oriented graph such that: $(x, y) \in F \cup F_r$ iff there exists an arc from $x$ to $y$.

We further define for all $x \in P \cup T$, the following sets: ${}^\bullet x = \{y \in P \cup T | (y, x) \in F\}$, and $x^\bullet = \{y \in P \cup T | (x, y) \in F\}$. And for all $x \in T$: ${}^\circ x = \{y \in P | (y, x) \in F_r\}$ . Those definitions naturally extend by union to the subsets of $P \cup T$.

For any transition $t$, we say that the places in ${}^\bullet t$ are *consumed* by $t$, the places in ${}^\circ t$ are *read* by $t$, and the places in $t^\bullet$ are *produced* by $t$.

For any place $p$, the transitions (and the corresponding arcs) in ${}^\bullet p$ are said to be *input* transitions (or arcs) of $p$ and those in $p^\bullet$ are said to be output transitions (or arcs) of $p$.

We suppose w.l.o.g. that for all transitions $t$, ${}^\bullet t \neq \emptyset$. If some transition $t$ consumes no place, we can simply add a new place $p$ in $P$ s.t. $(p, t) \in F$ and $(t, p) \in F$ and add $p$ in $m_0$. The obtained net is completely equivalent to the original net, even in the case of time or stopwatch Petri nets.

A *marking* of the net is a subset of $P$. We say that in the marking $m$, $p \in P$ contains a *token* if $p \in m$. A transition $t \in T$ is *enabled* by marking $m$ if ${}^\bullet e \cup {}^\circ e \subseteq m$. We denote by $\mathsf{Enabled}(m)$ the set of transitions enabled by marking $m$. A transition $t \in \mathsf{Enabled}(m)$ can be *fired*, leading to the new marking $m' = (m \setminus {}^\bullet e) \cup e^\bullet$. We denote this by $m \xrightarrow{e} m'$. A marking $m$ is *reachable* if there exists a finite sequence of transitions $t_1, \ldots, t_n$ such that $m_0 \xrightarrow{t_1} \cdots \xrightarrow{t_n} m$.

*Example 1* Figure 1 presents an example of Petri net. Places are depicted as circles, transitions as filled rectangles and the marking by the tokens in the places: here $m_0 = \{p_1, p_2\}$. There is a read arc betwen place $p_3$ and transition $t_4$ drawed using a dashed style.

Marking $\{p_3, p_6\}$ is reachable, for instance through the following sequence: $\{p_1, p_2\} \xrightarrow{t_1} \{p_2, p_3\} \xrightarrow{t_2} \{p_3, p_4\} \xrightarrow{t_0} \{p_1, p_2\} \xrightarrow{t_2} \{p_4, p_1\} \xrightarrow{t_1} \{p_3, p_4\} \xrightarrow{t_4} \{p_3, p_6\}$

Remark that transition $t_4$ is not fireable from $\{p_4, p_1\}$ because of the read arc reading $p_3$.

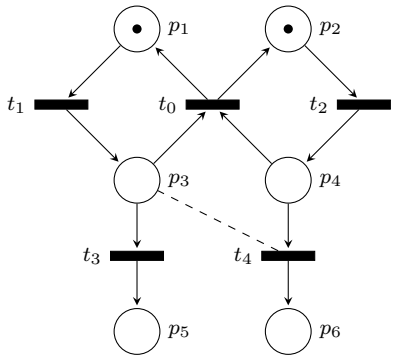


**Fig. 1** A Petri net

## 2.2 Branching Processes

We now present a concurrent semantics for Petri nets in terms of branching processes. Our definition adapts that of [13] for the accounting of read arcs.

Branching processes of a Petri net $\mathcal{N}$ are themselves Petri nets. In these Petri nets, places are called *conditions* and transitions are called *events*. Any condition is uniquely identified by the name $(p, \{x\})$ where $p$ is a place of $\mathcal{N}$ and $x$ is the name of the unique input event. Similarly, any event is uniquely identified by the name $(t, X, X')$ where $t$ is a transition of $\mathcal{N}$, $X$ is the set of the names of the consumed conditions and $X'$ is the set of the names of the read conditions.

For all conditions $b = (p, \{x\})$, we note $l(b) = p$. Similarly, for any event $e = (t, X, X')$, we note $l(e) = t$. Finally, for all markings $m$ of a branching process, we note $l(m)$ the marking of $\mathcal{N}$ such that $p \in m$ iff $l(p) \in l(m)$.

**Definition 2 (Branching Processes)** The set of *branching processes* of a Petri Net $\mathcal{N} = (P, T, F, F_r, m_0)$ is the smallest set of Petri nets satisfying the following conditions:

1. the Petri net without any transition and whose set of places and initial markings are $\{(p, \emptyset) | p \in m_0\}$ is a branching process of $\mathcal{N}$.

2. let $\mathcal{B}$ be a branching process of $\mathcal{N}$ and $m$ be a reachable marking of $\mathcal{B}$ such that some transition $t$ of $\mathcal{N}$ is fireable from marking $l(m)$ of $\mathcal{N}$. Let $M = \{p \in m | l(p) \in {}^\bullet t\}$ and $M' = \{p \in m | l(p) \in {}^\circ t\}$. $M$ (resp. $M'$) is a set of conditions corresponding to places consumed (resp. read) by $t$. Let $M'' = \{(p, \{(t, M, M')\}) | p \in t^\bullet\}$. The Petri net obtained by adding to $\mathcal{B}$ the conditions in $M''$ and the event $(t, M, M')$ with read arcs from $M'$, input arcs from $M$ and output arcs to $M''$ is also a branching process of $\mathcal{N}$. Event $(t, M, M')$ is called a *possible extension* of $\mathcal{B}$.
3. If $((B_i, E_i, F_i, F_{ri}, m_{0i}))_i$ is a finite or infinite family of branching processes of $\mathcal{N}$ then so is $(\bigcup_i B_i, \bigcup_i E_i, \bigcup_i F_i, \bigcup_i F_{ri}, \bigcup_i m_{0i})$.

*Example 2* Figure 2 shows a branching process of the Petri net in Figure 1. We have omitted the initial marking that is common to all branching processes. It is only useful for the formal definition.

Shorthands have been defined for the names of conditions $(b_1, b_2, \ldots)$ and events $(e_1, e_2, \ldots)$ to simplify the picture. The complete unique name of $e_5$, for instance, is:

$$\Big(t_4, \big\{(p_4, \{(t_2, \{(p_2, \emptyset)\}, \emptyset)\})\big\}, \big\{(p_3, \{(t_1, \{(p_1, \emptyset)\}, \emptyset)\})\big\}\Big)$$

We therefore have $l(e_5) = t_4$.

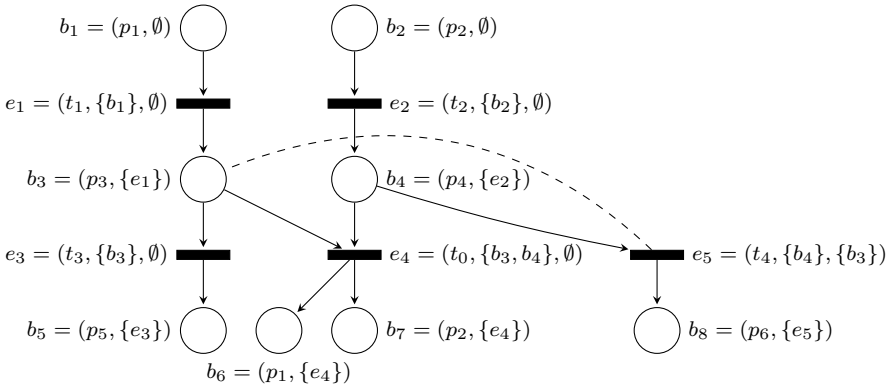


**Fig. 2** A branching process of the Petri net in Figure 1

This leads us to the definitions of unfolding and prefix:

**Definition 3 (Unfolding of a Petri net)** The *unfolding* $\mathsf{Unfolding}(\mathcal{N})$ of a Petri net $\mathcal{N}$ is the branching process obtained as the union of all branching processes of $\mathcal{N}$.

**Definition 4 (Prefix)** A branching process $\mathcal{B}$ is a *prefix* of another branching process $\mathcal{B}'$ if the set of events of $\mathcal{B}$ is included in that of $\mathcal{B}'$.

Let $\mathcal{B} = (B, E, F, F_r, m_0)$ be a branching process of a Petri net $\mathcal{N}$. We can relate the events in $E$ in several ways, the most fundamental one is causality.

**Definition 5 (Causality)** *Causality* is a partial order on $B \cup E$, denoted by $<$, and defined by: $x < y$ iff there exists a path in $\mathcal{B}$ from $x$ to $y$, with at least one arc in $F$. We note $x \leq y$ when $x = y$ or $x < y$.

We deduce from this the notion of causal history:

**Definition 6 (Causal history)** The *causal history* of an event $e \in E$ is $\lceil e \rceil = \{e' \in E | e' \leq e\}$.

The causal history of $e$ contains all the events required for the occurrence of $e$. For any subet $E' \subseteq E$, we define $\lceil E' \rceil = \bigcup_{e \in E'} \lceil e \rceil$.

*Example 3* In the branching process in Figure 2, the causal history of $e_3$ is $\lceil e_3 \rceil = \{e_1, e_3\}$ and that of $e_5$ is $\lceil e_5 \rceil = \{e_1, e_2, e_5\}$.

However, in a same branching process, two events may not belong to the same history. The simplest example of such a situation consists in two events consuming the same condition. This is called a conflict.

**Definition 7 (Conflicting event set)** A set $E'$ of events is conflicting, denoted by $\#E'$, if:

- $\exists e_1, e_2 \in \lceil E' \rceil$ s.t. $e_1 \neq e_2$ and ${}^\bullet e_1 \cap {}^\bullet e_2 \neq \emptyset$;
- or $\exists e_1, e_2, \ldots, e_n \in \lceil E' \rceil$ s.t. $e_1 = e_n$, $\exists i, j$ s.t. $e_i \neq e_j$ and $\forall k \in [1..n-1], e_k < e_{k+1}$ or ${}^\circ e_k \cap {}^\bullet e_{k+1} \neq \emptyset$.

Definition 7 is based on two notions: direct conflict and weak causality. Before we formally define those, we need the notion of co-set.

**Definition 8 (co-set)** A co-*set* of $\mathcal{B}$ is a set $B' \subseteq B$ of conditions that are concurrent, that is to say without causal relation nor conflict: $\forall b \in B', b^\bullet \cap \lceil {}^\bullet B' \rceil = \emptyset$ and not $\#\lceil {}^\bullet B' \rceil$.

Now, the first type of conflict occurs when to events consume the same common condition: we call that a direct conflict.

**Definition 9 (Direct conflict)** Two events $e_1, e_2 \in E$ are in direct conflict, denoted by $e_1 \,\mathsf{conf}\, e_2$ iff $e_1 \neq e_2$, ${}^\bullet e_1 \cup {}^\circ e_1 \cup {}^\bullet e_2 \cup {}^\circ e_2$ is a co-set and ${}^\bullet e_1 \cap {}^\bullet e_2 \neq \emptyset$.

In addition to the $<$ relation, read arcs induce another type of causality in branching processes: if some event should read a condition that is also consumed by another event, then the read must happen before the consumption.

**Definition 10 (Weak Causality)** *Weak causality* is the relation on events $\nearrow$ defined by: $e_1 \nearrow e_2$ iff $e_1 < e_2$ or ${}^\bullet e_1 \cup {}^\circ e_1 \cup {}^\bullet e_2 \cup {}^\circ e_2$ is a co-set and ${}^\circ e_1 \cap {}^\bullet e_2 \neq \emptyset$.

$e_1 \nearrow e_2$ implies that if $e_1$ occurs then it does before $e_2$.

*Example 4* In the branching process in Figure 2, even if $e_5$ does not belong to $\lceil e_3 \rceil$, we have $e_5 \nearrow e_3$ because ${}^\circ e_5 \cap {}^\bullet e_3 \neq \emptyset$. If those events both occur in the same history then $e_5$ does so before $e_3$.

Consequently, a second type of conflict comes from cycles in the weak causality relation, which corresponds to the second item in Definition 7. For instance, $e$ reads a condition consumed by $e'$ and $e'$ reads a condition consumed by $e$: in this case, only one of $e$ and $e'$ can occur in a given history. For $X \subseteq E$, we note $\mathsf{RCycles}(X)$ the set of event sets $\{e_1, e_2, \ldots, e_n\} \subseteq X$ s.t. $e_1 \nearrow e_2 \nearrow \cdots \nearrow e_n \nearrow e_1$ and ${}^\bullet\{e_1, \ldots, e_n\}$ is a $\mathsf{co}$-set.

*Example 5* In the branching process in Figure 2, $e_4 \,\mathsf{conf}\, e_5$ and therefore those two events are mutually exclusive.

In the branching process in Figure 3, we have the cycle $e_1 \nearrow e_2 \nearrow e_3 \nearrow e_1$. So these three events cannot coexist. They can pairwise however: $e_1$ an $e_2$ can coexist, as well as $e_1$ and $e_3$ or $e_2$ and $e_3$.

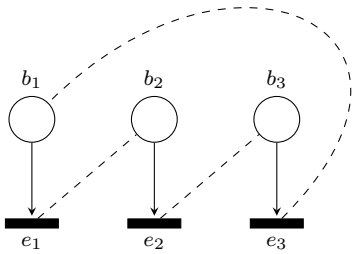


**Fig. 3** Weak causality cycle

Two events $e_1$ and $e_2$ such that ${}^\bullet e_1 \cap {}^\bullet e_2 \neq \emptyset$ can be not in direct conflict if some of their preconditions are themselves in conflict. The same holds for cycles of weak causality. We have the following result however:

**Lemma 1** *Let $E'$ be a set of events. $\#E'$ iff*

- $\exists e_1, e_2 \in \lceil E' \rceil$ *s.t.* $e_1 \,\mathsf{conf}\, e_2$*;*
- *or* $\mathsf{RCycles}(\lceil E' \rceil) \neq \emptyset$*.*

*Proof* The right-to-left direction is trivial so we focus on the left-to-right direction. Suppose then that $\#E'$. If one of the two conditions of Definition 7 is

satisfied and the preconditions $B$ of the corresponding events are not in conflict then we have the expected result. So suppose they are in conflict. Then we have a set $E'' = {}^\bullet B$ such that $\#E''$ and $E'' \subset E'$. We can apply the same reasoning to $E''$. Since $\subset$ on sets of events is well-founded, by definition of branching processes, we cannot repeat this reasoning an infinite number of times and thus the result holds. □

The possible behaviors of the original Petri net are obtained by deciding conflicts in the branching processes, which gives the notion of configuration, sometimes also called process.

**Definition 11 (Configuration)** A *configuration* of $\mathcal{B}$ is a subset $C \subseteq E$ of events that is *closed by causality* and *conflict-free*, that is to say: $\forall e' \in C, \forall e \in E,\ e < e' \Rightarrow e \in C$ and not $\#C$.

We denote by $\mathsf{Config}(\mathcal{B})$ the set of configurations of $\mathcal{B}$.

Remark, that for any event $e$, its causal history is a configuration. For all configurations $C$ and all possible extensions $(t, M, M')$ of $\mathcal{B}$, we note $C \cup \{(t, M, M')\}$ the set of events obtained by adding $(t, M, M')$ to $C$. We say that $(t, M, M')$ is a possible extension of $C$ if $C \cup \{(t, M, M')\}$ is again a configuration.

**Definition 12 (Cut)** A *cut* is a maximal $\mathsf{co}$-set (inclusion-wise). With each configuration $C$, we can associate a cut denoted by $\mathsf{Cut}(C)$ by $\mathsf{Cut}(C) = (M_0 \cup C^\bullet) \setminus {}^\bullet C$ (where $M_0$ is the marking common to all branching processes).

Note that if $C$ is finite then $l(\mathsf{Cut}(C))$ is the marking of the original Petri net after the sequence of transitions corresponding to the events in $C$. If $C$ is infinite it is only a partial marking consisting of the marked places not involved infinitely often in the firing of transition corresponding to the events in $C$.

*Example 6* In the branching process in Figure 2, the configurations are the causal histories of all events plus $\emptyset$, $\{e_1, e_2\}$ and $\{e_1, e_2, e_3\}$.

The cuts correspond to the application of $\mathsf{Cut}$ to all these configurations, which includes $\{b_3, b_8\} = \mathsf{Cut}(\lceil e_5 \rceil)$ and $\{b_1, b_2\} = \mathsf{Cut}(\emptyset)$.

The $\mathsf{co}$-sets are the non-empty subsets of these cuts, here all reduced to singletons.

## 3 Time Petri Nets and Time Branching Processes

We note $\mathcal{I}(\mathbb{Q}_{\geq 0})$ the set of intervals in $\mathbb{R}$ whose bounds are in $\mathbb{Q}_{\geq 0} \cup \{+\infty\}$. Let $I \in \mathcal{I}(\mathbb{Q}_{\geq 0})$. We denote by $I^\uparrow$ the *upward-closure* of $I$, that is to say the smallest interval of $\mathbb{R}$ that is right-open and right-unbounded and contains $I$. Similarly, we denote by $I^\downarrow$ the *downward-closure* of $I$, that is to say the smallest interval of $\mathbb{R}$ that is left-open and left-unbounded and contains $I$. For any interval $I$ and a real number $x$, we denote by $I + x$ the interval $\{y + x \in \mathbb{R} | y \in I\}$.

**Definition 13 (Time Petri Net)** A safe *time Petri net* with read arcs (TPN for short) is a tuple $\mathcal{N} = (P, T, F, F_r, m_0, I_s)$ such that:

- $(P, T, F, F_r, m_0)$ is a Petri net that we denote by $\mathsf{Untimed}(\mathcal{N})$;
- $I_s : T \to \mathcal{I}(\mathbb{Q}_{\geq 0})$ associates with each transition a *time constraint* as an interval called *static firing interval*.

For any transition $t$, the interval $I_s(t)$ constrains the dates at which $t$ can be fired: $t$ should be continuously enabled for a duration belonging to $I_s(t)$ and this duration must always belong to $I_s(t)^{\downarrow}$.

*Example 7* Figure 4 gives a "timed" version of the Petri net in Figure 1. Each transition now has a static firing time interval.

For instance, transition $t_1$ has no constraint on its firing dates while $t_2$ can only fire when at least 3 time units have passed since the arrival of a token in $p_2$ and it must fire strictly before 4 time units. Transition $t_0$ must fire immediately when there are tokens in both $p_3$ and $p_4$ (unless it is disabled in 0 time units by the firing of $t_3$ or $t_4$).

As we will see, for a given transition, other constraints, related to conflicts, must be considered in addition to the static firing time interval to find the actual occurrence dates of events.

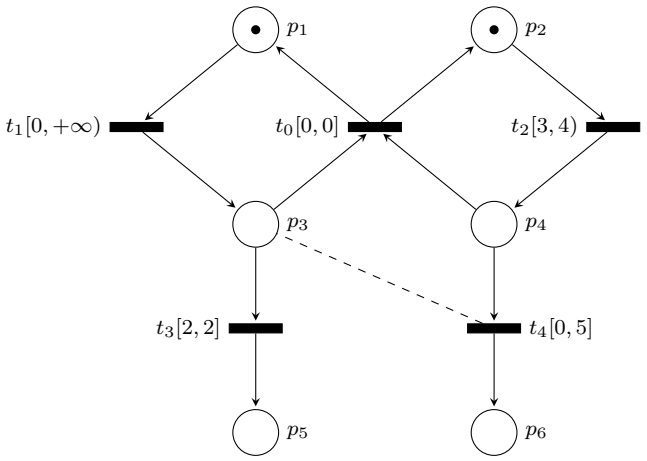


**Fig. 4** A Time Petri Net

We could give the semantics of time Petri nets as a "flat" timed transition system in which states consist of a marking and dynamic intervals for all transitions [4] or implicit clocks as in [21,16]. But since we are here interested in concurrency, we rather use the notion of time processes from [2].

**Definition 14 (Time process)** A *time process* of a TPN $\mathcal{N}$ is a pair $(C, \theta)$ such that:

- $C$ is a configuration (of a branching process) of $\mathsf{Untimed}(\mathcal{N})$;
- $\theta : C \to \mathbb{R}_{\geq 0}$ is timing function giving for each event in $C$ an *occurrence date*.

Of course, all possible values $\theta$ are not allowed by the time constraints in the TPN. The notion of *validity* of a timing function defines those time processes that do respect the time constraints. Before coming to the definition we need some more notations.

For a time process $(C, \theta)$ and $e \in C$, we define $\mathsf{Earlier}((C, \theta), e) = \{e' \in C | \theta(e') < \theta(e)\}$ the set of all events that have happened at an earlier date than $e$ in $(C, \theta)$. When the time process is clear from the contex, we note this only $\mathsf{Earlier}(e)$. Remark that $\mathsf{Earlier}(e)$ is always a configuration [2].

Let $t$ be a transition of $\mathcal{N}$ and $B' \subseteq C$ a $\mathsf{co}$-set such that $t$ is enabled by $l(B')$. We define the *date of enabling* of $t$ by $B'$ as: $\mathsf{TOE}(B', t) = \max(\{\theta(^\bullet b) | b \in B' \setminus M_0 \text{ and } l(b) \in {}^\bullet t \cup {}^\circ t\} \cup \{0\})$. That is to say that $\mathsf{TOE}$ is either 0 if all the read or consumed conditions belong to the initial marking, or the date of production of the most recent read or consumed condition otherwise.

**Definition 15 (Valid time process)** A time process $(C, \theta)$ of a TPN $\mathcal{N}$ is *valid* if for all $e \in C$,

$$\theta(e) - \mathsf{TOE}(^\bullet e \cup {}^\circ e, l(e)) \in I_s(l(e))^\uparrow \quad (1)$$

$$\forall t \in \mathsf{Enabled}(l(\mathsf{Cut}(\mathsf{Earlier}(e))), \theta(e) - \mathsf{TOE}(\mathsf{Cut}(\mathsf{Earlier}(e)), t) \in I_s(t)^\downarrow \quad (2)$$

If $(C, \theta)$ is a valid time process, we also say that $\theta$ is a valid timing function for $C$.

Eq. 1 states that the enabling duration should be passed the lower bound of the interval to fire, while Eq. 2 states that for *all* transitions enabled in the net by earlier non-consumed conditions should not go past the upper bound of their static firing time interval.

This last condition is a problem for unfoldings as it is not "local" to a given transition, adding constraints from other enabled transitions anywhere in the net.

For a time process $(C, \theta)$ and $C' \subseteq C$, we note, slightly abusively, $(C', \theta)$ the time process obtained with $C'$ and the restriction of $\theta$ to $C'$.

*Example 8* Figure 5 shows to valid time processes for the TPN in Figure 4. We recall the value of $l(e)$ besides each event $e$, then we give the occurrence date for $e$. The labels for events thus have the form $e; l(e); \theta(e)$. Similarly for conditions the labels have the form $b; l(b)$.

In the time process in Figure 5a, we could also have added $e_3$ at date 3.52 instead of $e_4$ (both events being in conflict).

In the time process in Figure 5b however, it is not possible to replace $e_3$ by $e_4$, whatever the date. $e_1$ occurring at date 0.13, $e_3$ is indeed forced to occur before $e_2$ which belongs to the causal history of $e_4$.

Similarly, in the time process in Figure 5a, for all occurrence dates of $e_2$ belonging to $[3, 3.52[$, the occurrence of $e_3$ is not possible. We therefore see that the occurrence of $e_3$ not only depends form its causal past $\{e_1\}$ but also from the occurrence of other events like $e_2$ that do not belong to it. This is due to the conflict between $e_3$ and $e_4$ which "transfers" some constraints from $e_4$ to $e_3$ (and the other way around).

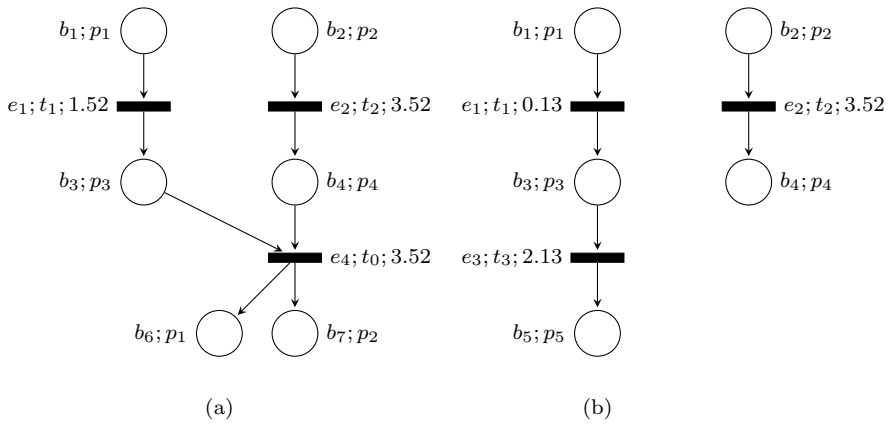


**Fig. 5** Two valid time processes of the TPN in Figure 4

## 4 Time Branching Processes

As we have just seen, time processes in a TPN are adding dates to *configurations* of the underlying Petri net and not to branching processes. The reason is that conflicting events are mutually exclusive and it is therefore difficult to give them simultaneously occurrence dates.

Our goal is however to add occurrence dates to branching processes so that we can define a timed unfolding in the same way as in the untimed case. For that purpose we extend the range of the timing function to include $+\infty$. This infinite value will be associated to events in the underlying branching process that do not occur due to timing constraints and conflicts. Using this extension of a timing function we define time branching processes:

**Definition 16 (Time branching process)** A *time branching process* (TBP for short) of a TPN $\mathcal{N}$ is a pair $(\mathcal{B}, \theta)$ where:

- $\mathcal{B} = (B, E, F, F_r, m_0)$ is a branching process of $\mathsf{Untimed}(\mathcal{N})$;
- $\theta : E \to \mathbb{R}_{\geq 0} \cup \{+\infty\}$ is a timing function giving for each event in $\mathcal{B}$ an occurrence date.

The definition of prefixes is naturally extended to TBPs:

**Definition 17** A TBP $(\mathcal{B}, \theta)$ is a prefix of a TBP $(\mathcal{B}', \theta')$ if $\mathcal{B}$ is a prefix of $\mathcal{B}'$ and the restriction of $\theta'$ to the events of $\mathcal{B}$ is equal to $\theta$.

For a set of events $E$ and a timing function $\theta$ on $E$, we note $E_{\theta<+\infty}$ the subset of events with a finite occurrence date defined by $E_{\theta<+\infty} = \{e \in E' | \theta(e) < +\infty\}$. By extension, for a TBP $(\mathcal{B}, \theta)$ with set of events $E$, we will write $\mathcal{B}_{\theta<+\infty}$ instead of $E_{\theta<+\infty}$.

Like for time processes, we need a number of constraints on a time branching process to ensure that the timing function respects the time constraints of the TPN. The first one is conflict-completeness.

**Definition 18 (conflict-completeness)** A TBP $(\mathcal{B}, \theta)$ is conflict-complete if for all possible extension $(t, M, M')$ of $\mathcal{B}$ such that $\forall b \in M \cup M', \theta(^\bullet b) \neq +\infty$, we have $(M \cup M') \cap {}^\bullet\mathcal{B}_{\theta<+\infty} = \emptyset$.

This definition intuitively means that when some event cannot occur due to an event in $E$ then it should also be in $\mathcal{B}$ (and as we will see, it will have an infinite occurrence date).

It is easy to make any TBP $(\mathcal{B}, \theta)$ conflict-complete by adding all the possible extensions contradicting the definition to $\mathcal{B}$ with an occurrence date $+\infty$. We note $\mathsf{ConfCompl}((\mathcal{B}, \theta))$ the resulting TBP.

We can now define the notion of valid time branching process.

**Definition 19 (Valid time branching process)** A time branching process $(\mathcal{B}, \theta)$ of a TPN $\mathcal{N}$, with $E$ the set of events of $\mathcal{B}$, is *valid* if it is conflict-complete and:

$$\forall E' \in \mathsf{RCycles}(E), \exists e' \in E' \text{ s.t. } \theta(e') = +\infty \tag{3}$$

and $\forall e \in E$ :

$$\Big[\big[\theta(e) \neq +\infty \text{ and } \theta(e) - \mathsf{TOE}(^\bullet e \cup {}^\circ e, l(e)) \in I_s(l(e)) \tag{4}$$

$$\text{and } \forall e' \in E \text{ s.t. } e \,\mathsf{conf}\, e', \theta(e') = +\infty \tag{5}$$

$$\text{and } \forall e' \in E \text{ s.t. } e \nearrow e', \theta(e) \leq \theta(e')\big] \tag{6}$$

$$\text{or } \big[\theta(e) = +\infty \text{ and } \exists b \in {}^\bullet e \cup {}^\circ e, \theta(^\bullet b) = +\infty\big] \tag{7}$$

$$\text{or } \big[\theta(e) = +\infty \text{ and } \exists e' \in E \text{ s.t. } (e \,\mathsf{conf}\, e' \text{ or } e \nearrow e') \text{ and } \theta(e') \neq +\infty$$
$$\text{and } \theta(e') - \mathsf{TOE}(^\bullet e \cup {}^\circ e, l(e)) \in I_s(l(e))^{\downarrow}\big]\Big] \tag{8}$$

Eq. 4 states that each transition should fire when it has been continuously enabled for a duration belonging to its static firing time interval.

Eq. 3 states that if there is a weak causality cycle, then at least one event in the cycle does not occur.

Eq. 5 states that if two events are in direct conflict then one of them does not occur.

Eq. 6 states that if there is a weak causality between two events, if the reading event occurs, then it is necessarily before the consuming event. The case when the causality is actually not weak is redundant with Eq. 4.

Eq. 7 states that is some event does not occur then none of its causal successor does.

Finally, equation 8 states that if some event does not occur, but all its causal successors do, then there exists another event in direct conflict (or that consumes a condition that had to be read) which has occurred before the first event was forced due to the upper bound of the static firing time interval. Since $(\mathcal{B}, \theta)$ should also be conflict-complete we are sure that these constraints are accounted for.

*Example 9* The set of constraints on the timing functions for the branching process in Figure 2, for the TPN in Figure 4, is given by:

- for $e_1$:
$$0 \leq \theta(e_1)$$
- for $e_2$:
$$3 \leq \theta(e_2) < 4$$
- for $e_3$:
$$\theta(e_3) = +\infty \text{ and } \theta(e_4) \leq \theta(e_1) + 2$$
$$\text{or}$$
$$\theta(e_3) = \theta(e_1) + 2 \text{ and } \theta(e_4) = +\infty$$
- for $e_4$:
$$\begin{cases} \theta(e_4) = +\infty \\ \text{and } (\theta(e_3) \leq \max\{\theta(e_1), \theta(e_2)\} \text{ or } \theta(e_5) \leq \max\{\theta(e_1), \theta(e_2)\}) \end{cases}$$
$$\text{or}$$
$$\begin{cases} \theta(e_4) = \max\{\theta(e_1), \theta(e_2)\} \\ \text{and } \theta(e_3) = +\infty \text{ and } \theta(e_5) = +\infty \end{cases}$$
- for $e_5$:
$$\begin{cases} \theta(e_5) = +\infty \\ \text{and } (\theta(e_3) \leq \max\{\theta(e_1), \theta(e_2)\} + 5 \text{ or } \theta(e_4) \leq \max\{\theta(e_1), \theta(e_2)\} + 5) \end{cases}$$
$$\text{or}$$
$$\begin{cases} \max\{\theta(e_1), \theta(e_2)\} \leq \theta(e_5) \leq \max\{\theta(e_1), \theta(e_2)\} + 5 \\ \text{and } \theta(e_4) = +\infty \text{ and } \theta(e_5) \leq \theta(e_3) \end{cases}$$

After simplification, we get:

$$\begin{cases} 1 \leq \theta(e_1) \\ 3 \leq \theta(e_2) < 4 \\ \theta(e_3) = +\infty \\ \theta(e_4) = \max\{\theta(e_1), \theta(e_2)\} \\ \theta(e_5) = +\infty \end{cases} \text{ or } \begin{cases} 0 \leq \theta(e_1) < 2 \\ 3 \leq \theta(e_2) < 4 \\ \theta(e_3) = \theta(e_1) + 2 \\ \theta(e_3) \leq \theta(e_2) \\ \theta(e_4) = +\infty \\ \theta(e_5) = +\infty \end{cases} \text{ or } \begin{cases} 0 \leq \theta(e_1) < 2 \\ 3 \leq \theta(e_2) < 4 \\ \theta(e_3) = \theta(e_1) + 2 \\ \theta(e_3) = \theta(e_2) \\ \theta(e_4) = +\infty \\ \theta(e_5) = \theta(e_2) \end{cases}$$

Among valid time branching processes we further distinguish those that are *temporally complete*. They are such that no "component" is too "late", that is to say that we cannot extend the branching process with an event that can only occur strictly before one that is already there.

**Definition 20 (Temporally complete TBP)** Let $(\mathcal{B}, \theta)$ be TBP of a TPN $\mathcal{N}$. $(\mathcal{B}, \theta)$ is *temporally complete* if for all extensions $(t, M, M')$ such that ${}^\bullet M \cup {}^\bullet M' \subseteq \mathcal{B}_{\theta<+\infty}$:

$$\max\{\theta(e) | e \in \mathcal{B}_{\theta<+\infty}\} - \mathsf{TOE}(M \cup M', t) \in I_s(t)^\downarrow$$

*Example 10* The time process in Figure 5b is also a time branching process. It is temporally complete. Its prefix obtained by removing $e_3$ (and condition $b_5$) is not however, because all the possible occurrence dates of $e_3$ (here the singleton $\{2.13\}$) are strictly less than that of $e_2$.

The correctness and completeness of the description of the behaviors of TPNs using time branching processes are established by Theorems 1 and 2 respectively. Those theorems will be proved as special cases of Theorems 5 and 6 in section 6.

**Theorem 1** *Let $\mathcal{N}$ be a TPN. Let $(\mathcal{B}, \theta)$ be temporally complete valid TBP of N.*

*$(\mathcal{B}_{\theta<+\infty}, \theta)$ is a valid time process of $\mathcal{N}$.*

**Theorem 2** *Let $\mathcal{N}$ be a TPN. Let $(C, \theta)$ be a valid time process of $\mathcal{N}$. $C$ defines a branching process $\mathcal{B} = (C^\bullet, C, F, F_r, m_0)$ of $\mathsf{Untimed}(\mathcal{N})$.*

*$\mathsf{ConfCompl}((\mathcal{B}, \theta))$ is a temporally complete valid TBP of $\mathcal{N}$.*

We can now define the notion of *symbolic unfolding* of a TPN in the same way we had defined the unfolding of an ordinary Petri net:

**Definition 21 (Symbolic unfolding)** The *symbolic unfolding* of a TPN $\mathcal{N}$ is the union of all its valid time branching processes.

## 5 Finite Complete Prefix

The unfolding of a Petri net is generally infinite. We can however always compute a finite prefix of this unfolding that is complete, i.e., that allows us to retrieve the whole unfolding [25]. We now show how the same thing can be done for TPNs using TBPs.

In this section, for the sake of simplicity, we consider TPNs without read arcs. Read arcs will be most useful for unfolding Stopwatch Petri Nets for which no finite complete prefix can exist (because reachability is undecidable). Read arcs also make the computation of a finite prefix tricky even for ordinary Petri nets [29].

In TBPs, the occurrence dates of events are always non-decreasing on causal paths but the behavior of the net actually only depends on the duration for which tokens have been produced:

**Definition 22 (Reduced age of a condition)** For any $\mathsf{co}$-set $A$, any timing function $\theta$ and any condition $b \in A$, the *reduced age* of $b$ in $A$ is:

$$\mathsf{age}(b, \theta, A) = \min\{\max_{b' \in A}\{\theta(^\bullet b')\} - \theta(^\bullet b), K^*(b)\}$$

$$\text{with } K^*(b) = \max\{K(t) \text{ s.t. } t \in T \text{ and } t \in l(b)^\bullet\}$$
$$\text{and } K(t) = \begin{cases} \max I_s(t) \text{ if } \max I_s(t) \neq +\infty \\ \min I_s(t) \text{ otherwise.} \end{cases}$$

**Definition 23 (Future-equivalent TBPs)** We say that two TBPs $(\mathcal{B}, \theta)$ and $(\mathcal{B}', \theta')$ are future-equivalent if:

- $l(\mathsf{Cut}(\mathcal{B}_{\theta<+\infty})) = l(\mathsf{Cut}(\mathcal{B}'_{\theta'<+\infty}))$,
- $\forall b \in l(\mathsf{Cut}(\mathcal{B}_{\theta<+\infty})), \forall b' \in l(\mathsf{Cut}(\mathcal{B}'_{\theta'<+\infty}))$ s.t. $l(b) = l(b')$: $\mathsf{age}(b, \theta, \mathsf{Cut}(\mathcal{B}_{\theta<+\infty})) = \mathsf{age}(b', \theta', \mathsf{Cut}(\mathcal{B}'_{\theta'<+\infty}))$.

**Proposition 1** *Let $(\mathcal{B}_1, \theta_1)$ and $(\mathcal{B}_2, \theta_2)$ be two future-equivalent TBPs. Let $(t, M_1)$ be an extension of $\mathcal{B}_1$. Then there exists an extension $(t, M_2)$ of $\mathcal{B}_2$.*

*Let $\delta_1 \in \mathbb{R}_{\geq 0}$. Let $\delta_2 = \delta_1 - \max_{b \in \mathsf{Cut}(\mathcal{B}_1)} \theta_1(^\bullet b) + \max_{b \in \mathsf{Cut}(\mathcal{B}_2)} \theta_2(^\bullet b)$. For $i \in \{1, 2\}$, let $(\mathcal{B}'_i, \theta'_i)$ be the TBP obtained by adding $(t, M_i)$ to $\mathcal{B}_i$ at date $\delta_i$.*

*If $\mathsf{ConfCompl}((\mathcal{B}'_1, \theta'_1))$ is valid then:*

- *$\mathsf{ConfCompl}((\mathcal{B}'_2, \theta'_2))$ is valid,*
- *$\mathsf{ConfCompl}((\mathcal{B}'_1, \theta'_1))$ and $\mathsf{ConfCompl}((\mathcal{B}'_2, \theta'_2))$ are future-equivalent,*

*Proof* We first prove that $\mathsf{ConfCompl}((\mathcal{B}'_2, \theta'_2))$ is valid. Since we do not consider read arcs, Eqs. 6 and 3 are irrelevant. We thus first prove that $e_2 = (t, M_2)$ satisfies Eqs. 4 and 5 in $(\mathcal{B}'_2, \theta'_2)$.

$\mathsf{ConfCompl}((\mathcal{B}'_1, \theta'_1))$ is valid so $e_1 = (t, M_1)$ satisfies those equations. Then $\theta'_1(e_1) - \mathsf{TOE}(^\bullet e_1, l(e_1)) \in I_s(l(e_1))$, or equivalently $\delta_1 - \max_{b \in M_1}\{\theta_1(^\bullet b)\} \in I_s(t)$. Also, $\delta_1$ being finite, it must be the case that $M_1 \subseteq \mathsf{Cut}(\mathcal{B}_1)$ and then we can choose $M_2 \subseteq \mathsf{Cut}(\mathcal{B}_{2\theta_2<+\infty})$. Since the reduced ages of the conditions are the same, we have $\delta_1 - \max_{b \in M_1}\{\theta_1(^\bullet b)\} = \delta_2 - \max_{b \in M_2}\{\theta_2(^\bullet b)\}$, and finally $\theta'_2(e_2) - \mathsf{TOE}(^\bullet e_2, l(e_2)) \in I_s(l(e_2))$. Moreover, since $M_2 \subseteq \mathsf{Cut}(\mathcal{B}_{2\theta_2<+\infty})$, any event $e'_2$ of $\mathcal{B}'_2$ such that $e_2 \,\mathsf{conf}\, e'_2$ must have an infinite occurrence date.

Eq. 7 is trivially satisfied because $(\mathcal{B}_2, \theta_2)$ is valid, $\delta_2$ is finite, and $\mathsf{ConfCompl}$ only adds extensions to conditions with a finite production date.

Finally, we need to prove Eq. 8. Let $e'_2$ be an event in $\mathcal{B}'_2$ such that $\theta'_2(e'_2) = +\infty$. If we do not have $e_2 \,\mathsf{conf}\, e'_2$ then Eq. 8 must be satisfied because $(\mathcal{B}_2, \theta_2)$ is valid. Else, if there exists some other event $e''_2$ in $\mathcal{B}_2$ such that $e''_2 \,\mathsf{conf}\, e'_2$, then again Eq. 8 must be satisfied because $(\mathcal{B}_2, \theta_2)$ is valid. Finally, if there is no such event, it must be the case that $^\bullet e'_2 \subseteq \mathsf{Cut}(\mathcal{B}_{2\theta_2<+\infty})$ and then $l(e'_2)$ is also possible from $\mathsf{Cut}(\mathcal{B}_{1\theta_1<+\infty})$ as an event $e'_1$ in direct conflict with $e_1$.

Since $\mathsf{ConfCompl}((\mathcal{B}'_1,\theta'_1))$ is valid and there is no other event in direct conflict, we have $\theta'_1(e_1) - \mathsf{TOE}({}^\bullet e'_1, l(e'_2)) \in I_s(l(e'_2))^\downarrow$. With the exact same reasoning as for Eq. 4, we get that Eq. 8 is satisfied for $e'_2$.

Let us now prove that $\mathsf{ConfCompl}((\mathcal{B}'_1,\theta'_1))$ and $\mathsf{ConfCompl}((\mathcal{B}'_2,\theta'_2))$ are future-equivalent. Since we fire the same transition in both processes, and since $l(\mathsf{Cut}(\mathcal{B}_{1\theta_1<+\infty})) = l(\mathsf{Cut}(\mathcal{B}_{2\theta_2<+\infty}))$, we immediately get $l(\mathsf{Cut}(\mathcal{B}'_{1\theta'_1<+\infty})) = l(\mathsf{Cut}(\mathcal{B}'_{2\theta'_2<+\infty}))$. So we now focus on the ages of conditions.

For $i \in \{1,2\}$, let $e_i^*$ be an event that realizes the maximum of finite occurrence dates in $\mathcal{B}_i$ and such that ${e_i^*}^\bullet \cap \mathsf{Cut}(\mathcal{B}_{i\theta_i<+\infty}) \neq \emptyset$. Then the conditions it produces that are in $\mathsf{Cut}(\mathcal{B}_{i\theta_i<+\infty})$ have age 0 in $(\mathcal{B}_i,\theta_i)$.

By definition of $\delta_2$, we then have: $\delta_2 = \delta_1 - \theta_1(e_1^*) + \theta_2(e_2^*)$.

Let $b_1$ be some condition in $\mathsf{Cut}(\mathcal{B}_{1\theta_1<+\infty})$ and $b_2$ the corresponding condition for $\mathcal{B}_2$. Since we have future equivalence for $(\mathcal{B}_1,\theta_1)$ and $(\mathcal{B}_2,\theta_2)$, then $\min\{\theta_1(e_1^*) - \theta_1({}^\bullet b_1), K^*(b_1)\} = \min\{\theta_2(e_2^*) - \theta_2({}^\bullet b_2), K^*(b_2)\}$. Note that since $l(b_1) = l(b_2)$, we have $K^*(b_1) = K^*(b_2)$.

Now let $b'_1$ be some condition in $\mathsf{Cut}(\mathcal{B}'_{1\theta'_1<+\infty})$ and $b'_2$ the corresponding condition for $\mathcal{B}'_2$.

First suppose that $\delta_1 > \theta_1(e_1^*)$. Then $\delta_2 > \theta_2(e_2^*)$. So if $b'_1 \in e_1^\bullet$ then $b'_2 \in e_2^\bullet$ and their ages are both zero. Otherwise, $b'_i \in \mathsf{Cut}(\mathcal{B}_{i\theta_i<+\infty})$, for $i \in \{1,2\}$ and:

- if $\delta_1 - \theta_1(e_1^*) \leq K^*(b'_1)$, the age of $b'_1$ in $(\mathcal{B}'_1,\theta'_1)$ is now $\delta_1 - \theta'_1({}^\bullet b'_1)$ which, with the above relations and since $\theta_i({}^\bullet b'_i) = \theta'_i({}^\bullet b'_i)$ for $i \in \{1,2\}$, is then equal to $\delta_2 - \theta'_2({}^\bullet b'_2)$;
- if $\delta_1 - \theta_1(e_1^*) > K^*(b'_1)$ then $\delta_2 - \theta_2(e_2^*) > K^*(b'_2)$. As a consequence, the reduced ages of both $b'_1$ and $b'_2$ are $K^*(b'_1) = K^*(b'_2)$.

Now suppose that $\delta_1 \leq \theta_1(e_1^*)$. Then $\delta_2 \leq \theta_1(e_2^*)$ and the ages of conditions not produced by $e_1$ and $e_2$ are unchanged so we consider only the case when $b'_1 \in e_1^\bullet$ and $b'_2 \in e_2^\bullet$:

- if $\theta_1(e_1^*) - \delta_1 \leq K^*(b'_1)$, the age of $b'_1$ in $(\mathcal{B}'_1,\theta_1)$ is then $\theta_1(e_1^*) - \delta_1$ which, as we have seen above, is equal to $\theta_2(e_2^*) - \delta_2$, i.e., the age of $b'_2$ in $(\mathcal{B}'_2,\theta'_2)$;
- if $\theta_1(e_1^*) - \delta_1 > K^*(b'_1)$, then the ages of both $b'_1$ and $b'_2$ are $K^*(b'_1) = K^*(b'_2)$. □

**Proposition 2** *Let $(\mathcal{B}_1,\theta_1)$ and $(\mathcal{B}_2,\theta_2)$ be two valid future-equivalent TBPs. If $(\mathcal{B}_1,\theta_1)$ is temporally complete then $(\mathcal{B}_1,\theta_1)$ is temporally complete.*

*Proof* Let $(\mathcal{B}_1,\theta_1)$ be a temporally complete valid TBP. And let $(\mathcal{B}_2,\theta_2)$ be a valid TBP future-equivalent to $(\mathcal{B}_1,\theta_1)$.

Let $(t, M_2, M'_2)$ be an extension of $\mathcal{B}_2$ such that ${}^\bullet M_2 \cup {}^\bullet M'_2 \subseteq \mathcal{B}_{2\theta_2<+\infty}$. We need to prove that $\max\{\theta_2(e) | e \in \mathcal{B}_{2\theta_2<+\infty}\} - \mathsf{TOE}(M_2 \cup M'_2, t) \in I_s(t)^\downarrow$.

First remark that since the two TBP are future equivalent, there exist some co-sets $M_1$ and $M'_1$ in $\mathcal{B}_1$ such that $(t, M_1, M'_1)$ is an extension of $\mathcal{B}_1$ and ${}^\bullet M_1 \cup {}^\bullet M'_1 \subseteq \mathcal{B}_{1\theta_1<+\infty}$.

Let $i \in \{1,2\}$.

Since $(\mathcal{B}_i,\theta_i)$ is valid, it is also conflict-complete, so $(M_i \cup M'_i) \cap {}^\bullet\mathcal{B}_{i\theta_i<+\infty} = \emptyset$ and therefore $(M_i \cup M'_i) \subseteq \mathsf{Cut}(\mathcal{B}_{i\theta_i<+\infty})$.

Let $e_i$ be an event such that $\theta_i(e_i) = \max\{\theta_i(e) | e \in \mathcal{B}_{i\theta_i<+\infty}\}$. We can suppose without loss of generality that $e_i^\bullet \cap \mathsf{Cut}(\mathcal{B}_{i\theta_i<+\infty}) \neq \emptyset$.

Then the (non-reduced) age of any condition $b_i \in M_i \cup M_i'$ is $\theta(e_i) - \theta({}^\bullet b_i)$ and since the TBPs are future-equivalent, there is a condition $b_{3-i} \in M_{3-i} \cup M_{3-i}'$ such that $l(b_i) = l(b_{3-i})$ and $\min\{\theta(e_i) - \theta({}^\bullet b_i), K^*(b_i)\} = \min\{\theta(e_{3-i}) - \theta({}^\bullet b_{3-i}), K^*(b_{3-i})\}$.

$\mathsf{TOE}(M_i \cup M_i', t)$ is actually the production date of one of the conditions in $M_i \cup M_i'$ and since $(\mathcal{B}_1, \theta_1)$ is temporally complete, we have $\max\{\theta_1(e) | e \in \mathcal{B}_{1\theta_1<+\infty}\} - \mathsf{TOE}(M_1 \cup M_1', t) \in I_s(t)^\downarrow$, i.e., the age of $\mathsf{TOE}(M_1 \cup M_1', t)$ is in $I_s(t)^\downarrow$.

So, if $I_s(t)^\downarrow \neq (-\infty, +\infty)$, then the non-reduced age of $\mathsf{TOE}(M_1 \cup M_1', t)$ is less or equal to $K^*(\mathsf{TOE}(M_1 \cup M_1', t))$ and therefore the reduced age of $\mathsf{TOE}(M_1 \cup M_1', t)$ is the same as its non-reduced age. By future-equivalence of the two TBPs, we have the same result for $\mathsf{TOE}(M_2 \cup M_2', t)$ and it also follows that $\max\{\theta_1(e) | e \in \mathcal{B}_{1\theta_1<+\infty}\} - \mathsf{TOE}(M_1 \cup M_1', t) = \max\{\theta_2(e) | e \in \mathcal{B}_{2\theta_2<+\infty}\} - \mathsf{TOE}(M_2 \cup M_2', t)$.

Finally, if $I_s(t)^\downarrow = (-\infty, +\infty)$, then the constraint we need to prove is trivially satisfied. □

We now define the events from which it is possible to stop the unfolding without losing information. For that we use the notion of *adequate order* from [24,14].

**Definition 24 (Adequate order)** A binary relation $\prec$ on the set of finite configurations is an *adequate order* if:

- $\prec$ is irreflexive and transitive;
- $\prec$ refine the prefix order: $C \subset C'$ implies $C \prec C'$;
- $\prec$ is preserved by extension: if $\mathsf{Cut}(C) = \mathsf{Cut}(C')$ then for all possible extensions $(t, M)$ of $C$ and the corresponding extension $(t, M')$ of $C'$, $C \prec C'$ implies $C \cup \{(t, M)\} \prec C' \cup \{(t, M')\}$.

Chatain et Khomenko have proved that any adequate order is well-founded [10]: any non-empty set of finite configurations admits smallest element for $\prec$.

Let $e$ be some event. We denote by $\mathcal{P}(e)$ the set of finite valid TBPs that contain $e$. Let $(\mathcal{B}, \theta)$ be a TBP in $\mathcal{P}(e)$. We note $\mathsf{Past}((\mathcal{B}, \theta), e)$ the smallest valid prefix of $(\mathcal{B}, \theta)$ that contains $\lceil e \rceil$. The finite occurence date events in this prefix are those that must have occurred for $e$ to occur itself. The set of such events is of course, inclusion-wise, greater or equal to $\lceil e \rceil$ and trivially less or equal to $\lceil e \rceil \cup \mathsf{Earlier}(e)$.

Finally, we note $\mathsf{Past}(e) = \bigcup_{(\mathcal{B},\theta) \in \mathcal{P}(e)} \mathsf{Past}((\mathcal{B}, \theta), e)$

**Definition 25 (Cut-off event)** Let $\mathcal{N}$ be a TPN and $\prec$ an adequate order on the configurations of $\mathcal{N}$.

An event $e$ of $\mathsf{Unfolding}(\mathsf{Untimed}(\mathcal{N}))$ is a *cut-off event* of the symbolic unfolding $\mathsf{Unfolding}(\mathcal{N})$ if for all $(\mathcal{B}, \theta) \in \mathsf{Past}(e)$, there exists another event $e'$ of $\mathsf{Unfolding}(\mathsf{Untimed}(\mathcal{N}))$ and $(\mathcal{B}', \theta') \in \mathsf{Past}(e')$ such that:

– $\lceil e' \rceil \prec \lceil e \rceil$;
– $l(\mathsf{Cut}(\lceil e' \rceil)) \prec l(\mathsf{Cut}(\lceil e \rceil))$;
– $(\mathcal{B}, \theta)$ and $(\mathcal{B}', \theta')$ are future-equivalent.

It is interesting to remark that when all transitions have a $[0,0]$ static firing interval or all have $[0,+\infty)$, Definition 25 degenerates to the first two conditions, i.e., the definition of [24,14] for ordinary Petri nets. In the $[0,0]$ case, for all events $e$, $\mathsf{Earlier}(e)$ is empty and then $\mathsf{Past}(e)$ is actually $\lceil e \rceil$ with all occurrence dates null. In the $[0,+\infty)$ case, the validity of TBPs imposes no constraint other than a $+\infty$ occurrence date in case of direct conflicts, so again $\mathsf{Past}(e)$ is actually $\lceil e \rceil$, but this time with unconstrained occurrence dates.

Using cut-off events we now define a prefix of the symbolic unfolding. Theorem 3 proves that it is complete and Theorem 4 that it is finite.

**Definition 26 (Maximal cut-off-free prefix)** Let $\mathcal{N}$ be a TPN. The *maximal cut-off-free prefix w.r.t* $\prec$ of $\mathsf{Unfolding}(\mathcal{N})$, denoted by $\mathit{CFP}_{\prec}(\mathcal{N})$ is, inclusion-wise, the greatest prefix of $\mathsf{Unfolding}(\mathcal{N})$ that contains no cut-off event.

**Theorem 3** *Let $\mathcal{N}$ be a TPN. For any finite valid time process $(C,\theta)$ of $\mathsf{Unfolding}(\mathcal{N})$, there exists a finite valid time process $(C',\theta')$ in $\mathit{CFP}_{\prec}(\mathcal{N})$ such that $(C,\theta)$ and $(C',\theta')$ are future-equivalent.*

*Proof* We adapt the proof scheme of [24,14].

Let $(C,\theta)$ be a finite valid time process of $\mathsf{Unfolding}(\mathcal{N})$. If $(C,\theta)$ belongs to $\mathit{CFP}_{\prec}(\mathcal{N})$ we are done so suppose that $(C,\theta)$ does not belong to $\mathit{CFP}_{\prec}(\mathcal{N})$. Then a cut-off event $e$ exists in $C$. Let $(\mathcal{B},\theta) = \mathsf{Past}(\mathsf{ConfCompl}((C,\theta)), e)$. By definition, $(\mathcal{B},\theta)$ is valid. Since $e$ is a cut-off event, there also exists another event $e'$ and a time process $(\mathcal{B}',\theta') \in \mathsf{Past}(e')$ such that $\lceil e' \rceil \prec \lceil e \rceil$, $l(\mathsf{Cut}(\lceil e' \rceil)) = l(\mathsf{Cut}(\lceil e \rceil))$, and $(\mathcal{B},\theta)$ and $(\mathcal{B}',\theta')$ are future-equivalent. Since $(\mathcal{B},\theta)$ is valid and can be extended into $\mathsf{ConfCompl}((C,\theta))$ then, by Proposition 1, $(\mathcal{B}',\theta')$ can also be extended to a valid TBP $(\mathcal{B}'',\theta'')$ that is future-equivalent to $\mathsf{ConfCompl}((C,\theta))$.

We furthermore have $l(\mathsf{Cut}(\lceil e' \rceil)) = l(\mathsf{Cut}(\lceil e \rceil))$, so $\lceil e' \rceil$ and $\lceil e \rceil$ can respectively be extended to ${\mathcal{B}''}_{\theta'' < +\infty}$ and $C$ by adding events corresponding to the same transitions. Since $\prec$ is preserved by extension, and $\lceil e' \rceil \prec \lceil e \rceil$, we furthermore have ${\mathcal{B}''}_{\theta'' < +\infty} \prec C$.

Now $(C,\theta)$ is valid so, by Theorem 2, $\mathsf{ConfCompl}((C,\theta))$ is temporally complete and, using Proposition 2, so is $(\mathcal{B}'',\theta'')$. So $({\mathcal{B}''}_{\theta'' < +\infty},\theta'')$ is a valid time process. If $({\mathcal{B}''}_{\theta'' < +\infty},\theta'')$ is not in $\mathit{CFP}_{\prec}(\mathcal{N})$, then we can iterate this reasoning. We thus obtain a sequence $(C_1,\theta_1),\ldots,(C_n,\theta_n)$ of time processes that are all future-equivalent and such that $(C_1,\theta_1) = (C,\theta)$ and $C_n \prec \cdots \prec C_1$. Since $\prec$ is well-founded, this sequence is necessarily finite and therefore $(C_n,\theta_n)$ belongs to $\mathit{CFP}_{\prec}(\mathcal{N})$. □

**Theorem 4** *Let $\mathcal{N}$ be a TPN. $\mathit{CFP}_{\prec}(\mathcal{N})$ has a finite number of events.*

*Proof* Let $\mathcal{B}$ be a finite branching process of $\mathsf{Untimed}(\mathcal{N})$. For any timing function $\theta$ such that $(\mathcal{B},\theta)$ is valid, we note $M((\mathcal{B},\theta) = l(\mathsf{Cut}(\mathcal{B}_{\theta<+\infty}))$ and we let $A((\mathcal{B},\theta)) = (a_1,\ldots,a_p)$ be the vector of the ages of the conditions in $\mathsf{Cut}(\mathcal{B}_{\theta<+\infty})$ relatively to the maximal occurrence date of events in the TBP.

$M((\mathcal{B},\theta))$ can be represented by a vector of booleans of length equal to the number of places in the net, which we denote $p$ in the rest of this proof. So $M$ can take only finitely many values over the set of all valid TBPs.

The constraints on $\theta$ for $(\mathcal{B},\theta)$ to be valid only involve differences between occurrence dates and can therefore be represented by a finite union of Difference Bound Matrices (DBMs) [6,11] with rational bounds. Furthermore, the denominators of those bounds are bounded by the LCM of the finite bounds in static firing time intervals. So there are only a finite number of such DBMs.

We can moreover transform those constraints on occurrence dates into constraints on the reduced ages of the conditions in $\mathsf{Cut}(\mathcal{B}_{\theta<+\infty})$ and again this involves simple differences between dates. The result of this transformation is again a finite union of DBMs, which have rational bounds with bounded denominators [9]. So $A$ takes only finitely many values on the set of valid TBPs.

For any event $e$ in $\mathsf{Unfolding}(\mathcal{N})$, we denote by $P(e)$ the union of all pairs $(M((\mathcal{B},\theta),A((\mathcal{B},\theta)))$ for all $(\mathcal{B},\theta)$ in $\mathsf{Past}(e)$. Again, $P(e)$ takes only finitely many values on the set of events in the symbolic unfolding.

Suppose now that $\mathit{CFP}_{\prec}(\mathcal{N})$ is infinite. Then it necessarily contains an infinite causal sequence of events $e_1 < e_2 < \cdots < e_n < \cdots$. And among these events there must be two events $e$ and $e'$ such that $e' < e$, $l(\mathsf{Cut}(\lceil e\rceil)) = l(\mathsf{Cut}(\lceil e'\rceil))$ and $P(e) = P(e')$ So, by definition of $P$, for any valid TBP $(\mathcal{B},\theta)$ in $\mathsf{Past}(e)$, there exists a valid TBP $(\mathcal{B}',\theta')$ in $\mathsf{Past}(e')$ such that both TBPs are future-equivalent. Furthermore, $e' < e$ so $\lceil e'\rceil \subseteq \lceil e\rceil$. $\prec$ is an adequate order so it refines the prefix order and therefore $\lceil e'\rceil \prec \lceil e\rceil$.

$e$ is therefore a cut-off event, which contradicts the definition of $\mathit{CFP}_{\prec}(\mathcal{N})$. So $\mathit{CFP}_{\prec}(\mathcal{N})$ is finite. □

## 6 Parametric Stopwatch Petri Nets

*Stopwatches*, i.e., clocks whose evolution can be temporarily "frozen", allows the modeling of more complex systems. Their most prominent use is maybe the modeling of real-time preemptive schedulers: stopwatches model the progress of tasks, and are frozen when tasks are preempted.

We consider here the model introduced in [5] and based on activator arcs. Those arcs will correspond to additional read arcs in the unfolding.

We further extend our stopwatch with parameters in the static firing intervals [28]: each bound of the interval is defined as a *linear expression* on a finite set of parameters $V$, i.e., an expression generated by the grammar $\phi ::= q|x|q*\phi|\phi+\phi$, where $q \in \mathbb{Q}$ and $x \in V$.

A *parametric interval* on $V$ is then an interval whose lower (resp. upper) bound is either open in $-\infty$ (resp. $+\infty$) or a linear expression on $V$. We denote by $\mathcal{I}_p(V)$ the set of parametric intervals on $V$.

A valuation $v$ on $V$ is a mapping of $V$ to $R$. We denote by by $\mathbb{R}^V$ the set of valuations on $V$. For a linear expression $L$ and a valuation $v$ on $V$, we denote by $v(L)$ the real number obtained by replacing each parameter $x$ in $V$ by the value $v(x)$. Similarly, for a parametric interval $I$, we denote by $v(I)$ the interval of $\mathbb{R}$ obtained by applying $v$ to each of its non-infinite bounds.

**Definition 27 (Parametric Stopwatch Petri Net)** A safe *Parametric Stopwatch Petri Net* (PSwPN fo short) with activator arcs is a tuple $\mathcal{N} = (P, T, F, F_r, F_a, m_0, I_s, V_0)$ with:

- $(P, T, F, F_r, m_0)$ a Petri net;
- $F_a \subseteq P \times T$ is the *activation* relation.
- $I_s : T \to \mathcal{I}_p(V)$ maps a *parametric constraint* to each transition, defined as a parametric interval called *static firing time interval*;
- $V_0 \subseteq \mathbb{R}^V$ is a set of *valid* valuations such that for all transitions $t \in T$ and all valuations $v \in V_0$, $v(I_s(t)) \neq \emptyset$ and $v(I_s(t)) \subseteq \mathbb{R}_{\geq 0}$.

For any valuation $v \in V_0$, we denote by $v(\mathcal{N})$ the net obtained by replacing each parametric interval $I$ by $v(I)$. Since $v \in V_0$, $v(\mathcal{N})$ is a Stopwatch Petri Net (SwPN) as defined in [5].

As before, for all transitions $t \in T$, we denote by ${}^{\diamond}t = \{p \in P | (p,t) \in F_a\}$ the set of *activating* places of $t$.

For any marking $m$, a transition $t$ is *active* if ${}^{\bullet}t \cup {}^{\circ}t \cup {}^{\diamond}t \subseteq m$, i.e., if $t$ is enabled by $m$ and all its activating places are marked. We denote by $\mathsf{Active}(m)$ the set of transitions that are active in marking $m$.

Intuitively, an SwPN works like a TPN if we replace the notion of (continuous) enabling duration by that of *activity duration*, i.e., the duration during which the transition has been active and continuously enabled. The time interval associated with each transition constrains this duration. When an enabled transition $t$ is linked to a place $p$ by the activation relation, the activity duration only increase when $p$ is marked. $p$ is however not involved in the enabling of $t$. When $t$ is enabled but $p$ is not marked the activity duration of the transition is "frozen": it keeps its value but does not evolve although time globally progresses. When $t$ is not enabled its activity duration is zero.

*Example 11* Figure 6 shows an example of PSwPN. It has a unique activator arc between $p_2$ and $t_1$ distinguished by a lozenge end and two parameters $a$ and $b$. $V_0$ contains all valuations that give non-negative values to $a$ and $b$. Consider the valuation $v \in V_0$ such that $v(a) = 2$ and $v(b) = 2$. It defines a SwPN with static firing interval $[2, 4]$ for $t_2$.

In that SwPN, the activity duration of $t_1$ progresses only when $p_2$ is marked and $t_1$ may therefore not fire before at least the sequence $t_2, t_3, t_2, t_3$ has occurred. This sequence indeed allows, at most, the increase of the activity duration of $t_1$ by 8 time units. Up to 4 more time units can then pass before $t_2$ must fire again, which allows to reach the 10 time units required for $t_1$ to fire.

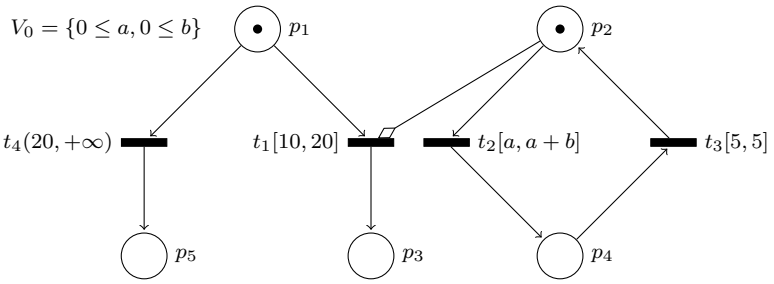


**Fig. 6** A parametric stopwatch Petri net

In TPNs, for a time process $(C, \theta)$, the enabling duration of a transition $t$ by the conditions in a co-set $B$ at date $\tau$ are given by $\tau - \mathsf{TOE}(B, t)$. It is this duration that is constrained by the notion of validity for time processes.

For SwPNs, the enabling duration is replaced by the activity duration in these constraints. We can compute this activity duration as the sum of all durations during which the transitions has been active since it was last enabled.

We now extend the notion of time process to PSwPNs, which allows us to give them a concurrent semantics.

For a PSwPN $\mathcal{N} = (P, T, F, F_r, F_a, m_0, I_s, V_0)$, we denote by $\mathsf{Underlying}(\mathcal{N})$ the Petri net with read arcs $(P, T, F, F_r \cup F_a, m_0)$. Remark that activator arcs of $\mathcal{N}$ are transformed into read arcs in $\mathsf{Underlying}(\mathcal{N})$ that express the dependency between a transition and its activating places.

We can now define time processes for PSwPNs:

**Definition 28 (Time process of an PSwPN)** A *time process* of a PSwPN $\mathcal{N}$ is a triple $(C, \theta, v)$ such that:

- $v$ is valid for $\mathcal{N}$;
- $C$ is a configuration of a branching process of $\mathsf{Underlying}(\mathcal{N})$;
- $\theta : C \to \mathbb{R}_{\geq 0}$ is a timing function giving for each event in $C$ an *occurrence date*.

Let $(C, \theta, v)$ be a time process of a PSwPN of $\mathcal{N}$.

The *duration* of a co-set $B$ in $(C, \theta)$, at date $\tau$ is:

$$\mathsf{Duration}(B, \tau) = \max\left\{0, \min\left\{\min_{e \in B^\bullet}\{\theta(e)\}, \tau\right\} - \max_{b \in B}\{\theta({}^\bullet b)\}\right\}$$

This is the date of the first consumption of one of the conditions in $B$ minus the date of the last production of one of the conditions in $B$. Between those two dates, the co-set exists: all of its conditions have been produced and none has been consumed yet.

For a transition $t$ and a co-set $B$ such that $l(B) = {}^\bullet t \cup {}^\circ t$, we denote $\mathsf{Acosets}(B, t)$ the set of co-sets $B \cup B'$ such that $l(B') = {}^\diamond t$.

The *activity duration* at date $\tau$ of a transition $t$ by the co-set $B$ is then:

$$\mathsf{DOA}(B,t,\tau) = \sum_{A \in \mathsf{Acosets}(B,t)} \mathsf{Duration}(A,\tau)$$

For any co-set $A$, we denote $\mathsf{En}(A,t) = \{b \in A | l(b) \in {}^\bullet t \cup {}^\circ t\}$ the set of conditions in $A$ that enable $t$ and ${}^\star e = \mathsf{En}({}^\bullet e \cup {}^\circ e, l(e))$, the conditions consumed or read by $e$ and that do not correspond to activating places.

If there are no activator arcs, remark that for any event $e$ we have ${}^\star e = {}^\bullet e \cup {}^\circ e$ and, as expected, $\mathsf{DOA}({}^\star e, l(e), \theta(e)) = \theta(e) - \mathsf{TOE}({}^\bullet e \cup {}^\circ e, t)$.

The validity of a time process is then given by:

**Definition 29 (Valid time process of a PSwPN)** A time process $(C,\theta,v)$ of an PSwPN $\mathcal{N}$ is *valid* if for all $e \in C$,

$$\theta(e) \geq \max_{b \in {}^\bullet e \cup {}^\circ e} \theta({}^\bullet b) \tag{9}$$

$$\mathsf{DOA}({}^\star e, l(e), \theta(e)) \in v(I_s(l(e)))^\uparrow \tag{10}$$

And $\forall t \in \mathsf{Active}(l(\mathsf{Cut}(\mathsf{Earlier}(e)))$:

$$\mathsf{DOA}(\mathsf{En}(\mathsf{Cut}(\mathsf{Earlier}(e)),t),t,\theta(e)) \in v(I_s(t))^\downarrow \tag{11}$$

We now have to enforce time progress with Eq. 9, since Eq. 10 is not enough for that. With that new equation, the following lemma trivially extends the one given in [2] for TPNs and states that the timing function for valid time processes is compatible with causality: if some event is a consequence of another one, then it does not occur before that event.

**Lemma 2** *Let $(C,\theta,v)$ be a valid TBP of a PSwPN.*

$$\forall e, e' \in C, e < e' \Rightarrow \theta(e) \leq \theta(e')$$

We finally prove the following Lemma for time processes that will be useful for the subsequent proofs.

**Lemma 3** *Let $(C,\theta,v)$ be a valid time process of a PSwPN. Let $t$ be a transition that is made active by exactly the set of conditions $A$ produced by events in $C$. Then for all $e \in {}^\bullet A$:*

- *either* $\mathsf{DOA}(\mathsf{En}(A,t),t,\theta(e)) = 0$,
- *or* $\exists e' \in C$ *such that* $\mathsf{DOA}(\mathsf{En}(A,t),t,\theta(e)) = \mathsf{DOA}(\mathsf{En}(A,t),t,\theta(e'))$ *and* $t \in \mathsf{Active}(l(\mathsf{Cut}(\mathsf{Earlier}(e'))))$.

*Proof* First remark that, since there is no conflict in $C$, if $B_0, B_1, \ldots, B_n, \ldots$ are the co-sets in $\mathsf{Acosets}(\mathsf{En}(A,t),t)$, then up to some renumbering, we have $B_0 < B_1 < \cdots < B_n < \cdots$, with $B_i < B_j$ iff $\lceil {}^\bullet B_i \rceil \subset \lceil {}^\bullet B_j \rceil$. Also, by definition, we have $\forall i, \mathsf{En}(B_i,t) = \mathsf{En}(A,t)$. Suppose $A = B_n$ and let $e_n \in {}^\bullet B_n$. We then have $\mathsf{Duration}(B_n, \theta(e_n)) = 0$ and:

- if $n = 0$ then $\mathsf{DOA}(\mathsf{En}(B_n, t), t, \theta(e_n)) = 0$;
- otherwise let $e'_{n-1}$ be the earliest in both the temporal and causal sense (one of them actually, because several can occur at the same date) event that consumes one of the activating conditions in $B_{n-1}$. Then by definition of $\mathsf{DOA}$, since $\mathsf{Duration}(B_n, \theta(e_n)) = 0$, we have $\mathsf{DOA}(\mathsf{En}(B_n, t), t, \theta(e_n)) = \mathsf{DOA}(\mathsf{En}(B_{n-1}, t), t, \theta(e'_{n-1}))$. There are now two further possibilities:
  - if $B_{n-1} \subseteq \mathsf{Cut}(\mathsf{Earlier}(e'_{n-1}))$, then we have the expected result because $\mathsf{En}(B_{n-1}, t) = \mathsf{En}(A, t)$.
  - otherwise, it must be the case that $\mathsf{Duration}(B_{n-1}, \theta(e'_{n-1})) = 0$ and there exists an event $e_{n-1} \in {}^\bullet B_{n-1}$ such that $\theta(e_{n-1}) = \theta(e'_{n-1})$. So we can repeat the above reasoning with $n-1$ instead of $n$ and at some point we will either have $B_n \subseteq \mathsf{Cut}(\mathsf{Earlier}(e_n))$ or $n = 0$. □

Time branching processes for PSwPNs are similarly obtained from the TPN case:

**Definition 30 (Time branching process of a PSwPN)** A *time branching process* of a PSwPN $\mathcal{N}$ is a pair $(\mathcal{B}, \theta, v)$ such that:

- $v$ is valid for $\mathcal{N}$;
- $\mathcal{B} = (B, E, F, F_r, m_0)$ is a branching process of $\mathsf{Underlying}(\mathcal{N})$;
- $\theta : E \to \mathbb{R}_{\geq 0} \cup \{+\infty\}$ is a timing function giving for each event in $E$ an *occurrence date*.

**Definition 31 (Valid time branching process of an SwPN)** A time branching process $(\mathcal{B}, \theta, v)$ of an SwPN $\mathcal{N}$, with set of events $E$, is *valid* if it is conflict-complete, $v$ is valid for $\mathcal{N}$, and:

$$\forall E' \in \mathsf{RCycles}(E), \exists e' \in E' \text{ s.t. } \theta(e') = +\infty \tag{12}$$

and $\forall e \in E$ :

$$\Big[\big[\theta(e) \neq +\infty \text{ and } \theta(e) \geq \max_{b \in {}^\bullet e \cup {}^\circ e} \theta({}^\bullet b) \tag{13}$$

$$\text{and } \mathsf{DOA}({}^\star e, l(e), \theta(e)) \in v(I_s(l(e))) \tag{14}$$

$$\text{and } \forall e' \in E \text{ s.t. } e \,\mathsf{conf}\, e', \theta(e') = +\infty \tag{15}$$

$$\text{and } \forall e' \in E \text{ s.t. } e \nearrow e', \theta(e) \leq \theta(e')\big] \tag{16}$$

$$\text{or } \big[\theta(e) = +\infty \text{ and } \exists b \in {}^\bullet e \cup {}^\circ e, \theta({}^\bullet b) = +\infty\big] \tag{17}$$

$$\text{or } \big[\theta(e) = +\infty \text{ and } \exists e' \in E \text{ s.t. } (e \,\mathsf{conf}\, e' \text{ or } e \nearrow e') \text{ and } \theta(e') \neq +\infty$$

$$\text{and } \mathsf{DOA}({}^\star e, l(e), \theta(e')) \in v(I_s(l(e)))^\downarrow\big]\Big] \tag{18}$$

Remark that $\mathsf{conf}$ and $\nearrow$ are interpreted on $\mathsf{Underlying}(\mathcal{N})$ and therefore also take the read arcs induced by the activator arcs into account.

*Example 12* Figure 7 shows a valid time branching process of the PSwPN in Figure 6 with $v(a) = 2$ and $v(b) = 2$.

Remark that, in particular, events $e_1$ and $e_4$ correspond to "missed" potential firings of $t_1$ because the token in $p_2$ has been consumed each time by $t_2$. The activity duration of $t_1$ was anyway less than 10 and therefore $t_1$ was not firable.

When $e_2$ occurs, making $t_1$ inactive, we have $\mathsf{DOA}(\{b_1\}, t_1, \theta(e_2)) = 3.2$, which is indeed less than 10. Similarly, when $e_5$ occurs, $\mathsf{DOA}(\{b_1\}, t_1, \theta(e_5)) = 11.3 - 8.2 + 3.2 - 0 = 6.3$.

These two events have read arcs in input that correspond to the activator arc between $t_1$ and $p_2$.

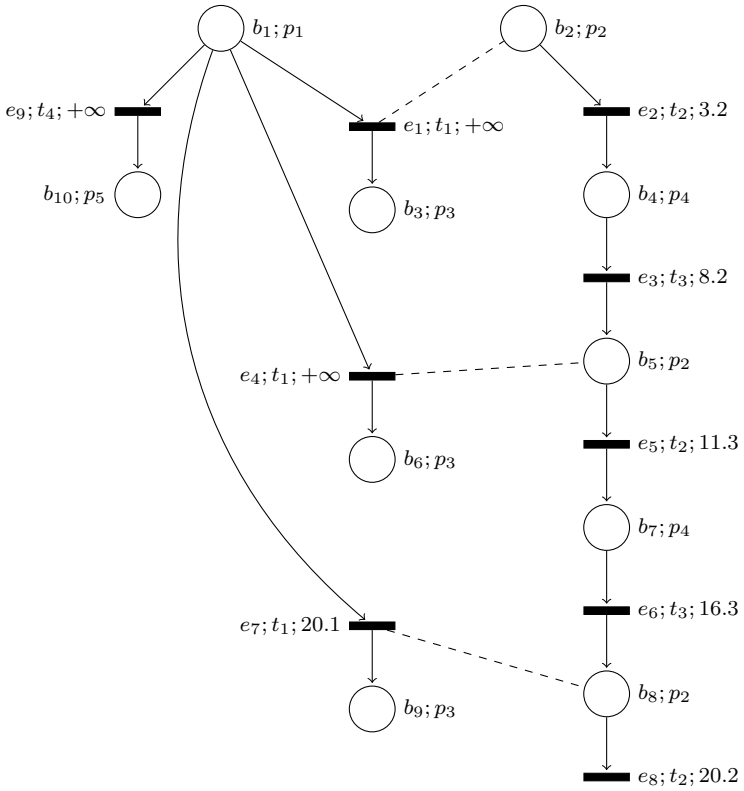


**Fig. 7** A valid time branching process of the net in Figure 6

Naturally, the notion of temporal completeness also needs to be updated to account for the stopwatches.

**Definition 32 (Temporally complete TBP of a PSwPN)** Let $(\mathcal{B}, \theta, v)$ be TBP of a PSwPN $\mathcal{N}$. $(\mathcal{B}, \theta)$ is *temporally complete* if for all extensions

$(t, M, M')$, such that ${}^\bullet M \cup {}^\bullet M' \subseteq E_{\theta<+\infty}$:

$$\mathsf{DOA}(\mathsf{En}(M \cup M', t), t, \max\{\theta(e) | e \in \mathcal{B}_{\theta<+\infty}\}) \in v(I_s(t))^\downarrow$$

The correctness and completeness of the time branching processes for PSwPNs is given by Theorems 5 and 6 but before we get to them, we need to establish a few lemmas: Lemma 4 lifts the compatibility with causality of Lemma 2 to timing functions of valid TBPs. Lemma 5 proves that the set of events that actually occur in a valid TBP forms a configuration. Finally, Lemma 6 makes explicit the rather obvious fact that any event consuming a condition produced and not consumed by the events having occurred strictly before some event $e$ may itself not have occured strictly before $e$.

**Lemma 4** *Let $(\mathcal{B}, \theta, v)$ be a valid TBP of a PSwPN and $E$ be its set of events.*

$$\forall e, e' \in E, e < e' \Rightarrow \theta(e) \leq \theta(e')$$

*Proof* Since $e < e'$, there exists a path from $e$ to $e'$, going through events $e_1, e_2, \ldots, e_n$ with $e_1 = e$ and $e_n = e'$. We moreover have for all $i \in [1..n-1]$, $e_i^\bullet \cap ({}^\bullet e_{i+1} \cup {}^\circ e_{i+1}) \neq \emptyset$. Since $(\mathcal{B}, \theta, v)$ is valid, if $\theta(e_{i+1}) \neq +\infty$ then, by Eq. 13, $\theta(e_i) \leq \theta(e_{i+1})$. If $\theta(e_{i+1}) = +\infty$ then we obviously also have $\theta(e_i) \leq \theta(e_{i+1})$. So by transitivity, $\theta(e) \leq \theta(e')$. □

**Lemma 5** *Let $(\mathcal{B}, \theta, v)$ be a valid TBP of a PSwPN and $E$ be its set of events. $E_{\theta<+\infty} = \{e \in E | \theta(e) < +\infty\}$ is a configuration of $\mathcal{B}$.*

*Proof* First, $E_{\theta<+\infty}$ is causally closed: let $e \in E_{\theta<+\infty}$ and let $e' < e$. Using lemma 4, we have $\theta(e') \leq \theta(e)$. Since $\theta(e) < +\infty$, by definition of $E_{\theta<+\infty}$, we have $\theta(e') < +\infty$ and finally $e' \in E_{\theta<+\infty}$.

Second, $E_{\theta<+\infty}$ is conflict-free: Suppose the opposite, i.e., $\#E_{\theta<+\infty}$. Then, by Lemma 1:

- either $\mathsf{RCycles}(E_{\theta<+\infty}) \neq \emptyset$, but since $(\mathcal{B}, \theta, v)$ is valid, this is forbidden by Eq. 12 that enforces that at least one of the events of such cycles has an infinite occurrence date;
- or $\exists x, y \in E_{\theta<+\infty}$ s.t. $x \,\mathsf{conf}\, y$. By definition of $E_{\theta<+\infty}$, we have $\theta(x') \neq +\infty$ and $\theta(y') \neq +\infty$ which contradicts Eq. 15. □

**Lemma 6** *Let $(\mathcal{B}, \theta, v)$ be a valid TBP of a PSwPN and $E$ be its set of events. Let $e \in E$ s.t. $\theta(e) \neq +\infty$. Then:*

$$\forall e' \in E, {}^\bullet e' \cap \mathsf{Cut}(\mathsf{Earlier}(e)) \neq \emptyset \Rightarrow \theta(e) \leq \theta(e')$$

*Proof* Suppose *a contrario* that $\theta(e') < \theta(e)$. Then $e' \in \mathsf{Earlier}(e)$. By definition of $\mathsf{Cut}$, each condition being produced exactly once, no condition in ${}^\bullet e'$ can belong to $\mathsf{Cut}(\mathsf{Earlier}(e))$, which contradicts ${}^\bullet e' \cap \mathsf{Cut}(\mathsf{Earlier}(e)) \neq \emptyset$. □

Now we can state and prove the following key theorems:

**Theorem 5** *Let $\mathcal{N}$ be a PSwPN. Let $(\mathcal{B},\theta,v)$ be temporally complete valid TBP of $\mathcal{N}$.*

*$(\mathcal{B}_{\theta<+\infty},\theta,v)$ is a valid time process of $\mathcal{N}$.*

*Proof* Lemma 5 already proves that $\mathcal{B}_{\theta<+\infty}$ is a configuration and therefore that $(\mathcal{B}_{\theta<+\infty},\theta,v)$ is a time process. We need to prove that this time process is valid.

Let $e \in \mathcal{B}_{\theta<+\infty}$. By definition, we have $\theta(e) \neq +\infty$.

Eq. 13 and Eq. 9 match. Eq. 14 immediately gives the satisfaction of the constraint in Eq. 10. Only $\mathsf{DOA}(\mathsf{En}(\mathsf{Cut}(\mathsf{Earlier}(e)),t),t,\theta(e)) \in v(I_s(t))^{\downarrow}$ for all $t \in \mathsf{Active}(l(\mathsf{Cut}(\mathsf{Earlier}(e)))$ remains to be proved.

Let $t \in \mathsf{Active}(l(\mathsf{Cut}(\mathsf{Earlier}(e)))$ and let $B' = \{b \in \mathsf{Cut}(\mathsf{Earlier}(e)) | l(b) \in {}^\bullet t \cup {}^\circ t \cup {}^\diamond t\}$ be the subset of $\mathsf{Cut}(\mathsf{Earlier}(e))$ that made $t$ active when it fired. Three cases may arise:

- if $t = l(e)$ then, the net being safe, $B' = {}^\bullet e \cup {}^\circ e$. Furthermore, by Eq. 14, $\mathsf{DOA}({}^\star e, l(e), \theta(e)) \in v(I_s(l(e)))^{\downarrow}$, i.e., $\mathsf{DOA}(\mathsf{En}(B',t),t,\theta(e)) \in v(I_s(t))^{\downarrow}$ and we find the expected constraint since $B' \subseteq \mathsf{Cut}(\mathsf{Earlier}(e))$.
- if $t \neq l(e)$ but there exists $e' \in E$ s.t. $l(e') = t$ and ${}^\bullet e' \cup {}^\circ e' = B'$ then:
  - If $\theta(e') \neq +\infty$ then, by Eq. 14, $\mathsf{DOA}(\mathsf{En}(B',t),t,\theta(e')) \in v(I_s(t))^{\downarrow}$. And with lemma 6, since ${}^\bullet e' \neq \emptyset$ then ${}^\bullet e' \cap \mathsf{Cut}(\mathsf{Earlier}(e)) \neq \emptyset$, and we have $\theta(e) \leq \theta(e')$. The expected constraint follows.
  - If $\theta(e') = +\infty$ then one of Eqs. 17 or 18 is satisfied. But since ${}^\bullet e' \cup {}^\circ e' = B' \subseteq \mathsf{Cut}(\mathsf{Earlier}(e))$, each of the conditions in $B'$ has a finite production date, which excludes Eq. 17. As a consequence, there exists $e'' \in E$ s.t. either $e' \,\mathsf{conf}\, e''$, or $e' \nearrow e''$, and $\theta(e'') \neq +\infty$ and $\mathsf{DOA}(\mathsf{En}(B',t),t,\theta(e'')) \in v(I_s(t))^{\downarrow}$. We cannot have $e' < e''$ because $\theta(e') = +\infty$ and that would contradict lemma 4. So either ${}^\bullet e' \cap {}^\bullet e'' \neq \emptyset$, or ${}^\circ e' \cap {}^\bullet e'' \neq \emptyset$. So ${}^\bullet e'' \cap \mathsf{Cut}(\mathsf{Earlier}(e)) \neq \emptyset$ and with lemma 6, $\theta(e) \leq \theta(e'')$, and the expected constraint again follows.
- if no event in $E$ correspond to the firing of $t$, then there must exist a possible extension $(t, M, M')$ of $\mathcal{B}$ s.t. $M \cup M' \subseteq \mathsf{Cut}(\mathsf{Earlier}(e))$. In that case, since $\mathcal{B}$ is temporally complete, by definition 32, we directly have $\mathsf{DOA}(\mathsf{En}(\mathsf{Cut}(\mathsf{Earlier}(e)),t),t,\theta(e)) \in v(I_s(t))^{\downarrow}$. □

**Theorem 6** *Let $\mathcal{N}$ be an SwPN. Let $(C,\theta,v)$ be a valid time process of $\mathcal{N}$. $C$ defines a branching process $\mathcal{B} = (C^\bullet, C, F, F_r, m_0)$ of $\mathsf{Untimed}(\mathcal{N})$.*

*$\mathsf{ConfCompl}((\mathcal{B},\theta,v))$ is a temporally complete valid TBP of $\mathcal{N}$.*

*Proof* Since $C$ is a configuration, we necessarily have $\mathsf{RCycles}(C) = \emptyset$ and since $\mathsf{ConfCompl}$ only adds events with an infinite occurrence date, Eq. 12 is satisfied.

Now let $e$ be some event in $\mathsf{ConfCompl}((\mathcal{B},\theta,v))$. First suppose that $e \in C$.

$(C,\theta,v)$ being a time process, we have $\theta(e) \neq +\infty$. So Eqs. 17 and 18 are satisfied by default.

Eqs. 9 and 13 match. Eq. 14 is trivially satisfied thanks to the constraints in Eqs. 10 and 11.

Suppose that there exists some $e' \in \mathsf{ConfCompl}((\mathcal{B}, \theta, v))$ such that $e \,\mathsf{conf}\, e'$. Then $e' \notin C$ because $C$ is a configuration. So $e'$ was added by $\mathsf{ConfCompl}$ and its occurrence date is $+\infty$ and satisfies Eq. 15.

For Eq. 16, let $e' \in \mathsf{ConfCompl}((\mathcal{B}, \theta))$ s.t. $e \nearrow e'$. If $\theta(e') = +\infty$ then for sure $\theta(e) \leq \theta(e')$. Else, for sure $e' \in C$. If $e < e'$ then, since $(C, \theta, v)$ is valid, lemma 2 gives us $\theta(e) \leq \theta(e')$. Else, ${}^\circ e \cap {}^\bullet e' \neq \emptyset$ and if $\theta(e') < \theta(e)$ then some condition in ${}^\bullet e$ are consumed strictly before $e$ occurs, which is impossible because the net is safe. So $\theta(e) \leq \theta(e')$.

Suppose now that $e \notin C$. Then, it has been added by $\mathsf{ConfCompl}$ and we have $\theta(e) = +\infty$ but all its preconditions have a finite production date. So again, Eq. 17 is satisfied by default. We also know that there must exist $e' \in C$ such that $e \,\mathsf{conf}\, e'$ or $e \nearrow e'$. We chose $e'$ as the earliest of such conflicting events. This is possible because $C$ is a configuration so there it has no conflict and therefore only a finite number of events in $C$ can consume the preconditions of $e$.

If $l(e) \in \mathsf{Active}(l(\mathsf{Cut}(\mathsf{Earlier}(e'))))$, Eq. 11 directly gives the expected Eq. 18. Otherwise:

- if there exists some condition $b \in {}^\bullet e \cup {}^\circ e$ such that $\theta(e') \leq \theta({}^\bullet b)$, then we can apply Lemma 3 to ${}^\bullet b$ and we get either $\mathsf{DOA}(\mathsf{En}({}^\bullet e \cup {}^\circ e, l(e)), l(e), \theta(e')) = 0$ or $\exists e'' \in C$ such that $\mathsf{DOA}(\mathsf{En}({}^\bullet e \cup {}^\circ e, l(e)), l(e), \theta(e')) \leq \mathsf{DOA}(\mathsf{En}({}^\bullet e \cup {}^\circ e, l(e)), l(e), \theta(e''))$ and $l(e) \in \mathsf{Active}(l(\mathsf{Cut}(\mathsf{Earlier}(e''))))$. Again Eq. 11 implies Eq. 18;
- otherwise ${}^\bullet e \cup {}^\circ e \subseteq \mathsf{Earlier}(e')^\bullet$. Since ${}^\bullet e \cup {}^\circ e \not\subseteq \mathsf{Cut}(\mathsf{Earlier}(e'))$, there exists some event $e'' \in \mathsf{Earlier}(e')$ and some condition $b \in {}^\bullet e \cup {}^\circ e$ such that $b \in {}^\bullet e''$, which is not possible due to the way $e'$ was chosen.

We finally prove that $(\mathcal{B}, \theta, v)$ is temporally complete. Let $(t, M, M')$ be a possible extension of $\mathcal{B}$ such that ${}^\bullet M \cup {}^\bullet M' \in C$. If there exists $e \in C$ such that $M \cup M' \subseteq \mathsf{Cut}(\mathsf{Earlier}(e))$ then Eq. 11 directly gives the expected constraint. Otherwise it means that the event $e$ that realizes the maximum of the finite occurrence dates in $C$ is actually in ${}^\bullet M \cup {}^\bullet M'$ and by Lemma 3 coupled with Eq. 11 we get the temporal completeness constraint. □

As for TPNs the notion of *symbolic unfolding* of a PSwPN easily follows:

**Definition 33 (Symbolic unfolding of an SwPN)** The *symbolic unfolding* of a PSwPN $\mathcal{N}$ is the union of all its valid time branching processes.

*Example 13* Let us consider the time branching process in Figure 7. For the sake of simplicity, we focus on the prefix obtained by removing $e_6, e_7$ and $e_8$. With all the constraints on the timing functions and the parameters this gives us a prefix of the symbolic unfolding. These constraints are:

- for $e_1$:

$$\theta(e_1) \in [10, 20] \text{ and } \theta(e_4) = +\infty \text{ and } \theta(e_9) = +\infty \text{ and } \theta(e_1) \leq \theta(e_2)$$
$$\text{or}$$
$$\theta(e_1) = +\infty \text{ and } (\theta(e_9) \leq 20 \text{ or } \theta(e_2) \leq 20)$$

– for $e_2$:

$$\theta(e_2) \in [a, a+b]$$

– for $e_3$:

$$\theta(e_3) = \theta(e_2) + 5$$

– for $e_4$:

$$\begin{cases} \big(\theta(e_4) \geq \theta(e_3) \text{ and } \theta(e_4) - \theta(e_3) + \theta(e_2) - 0 \in [10, 20] \\ \text{and } \theta(e_4) \leq \theta(e_5) \text{and } \theta(e_1) = +\infty \text{ and } \theta(e_9) = +\infty\big) \end{cases}$$
$$\text{or}$$
$$\begin{cases} \theta(e_4) = +\infty \\ \text{and } (\theta(e_9) - \theta(e_3) + \theta(e_2) \leq 20 \text{ or } \theta(e_5) - \theta(e_3) + \theta(e_2) \leq 20) \end{cases}$$

– for $e_5$:

$$\theta(e_5) \in [\theta(e_3) + a, \theta(e_3) + a + b]$$

– for $e_9$:

$$\theta(e_9) > 20 \text{ and } \theta(e_1) = +\infty \text{ and } \theta(e_4) = +\infty$$
$$\text{or}$$
$$\theta(e_9) = +\infty \text{ and } (\theta(e_1) \leq 20 \text{ or } \theta(e_4) \leq 20)$$

After a few simplifications, we get:

$$\begin{cases} \theta(e_1) = +\infty \\ \theta(e_2) \in [a, a+b] \\ \theta(e_2) \leq 20 \\ \theta(e_3) \in [5+a, 5+a+b] \\ \theta(e_4) = +\infty \\ \theta(e_5) \in [5+2a, 5+2a+2b] \\ \theta(e_9) \in (20, 25] \end{cases} \quad \text{or} \quad \begin{cases} \theta(e_1) = +\infty \\ \theta(e_2) \in [a, a+b] \\ \theta(e_2) \leq 20 \\ \theta(e_3) \in [5+a, 5+a+b] \\ \theta(e_4) = +\infty \\ \theta(e_5) \in [5+2a, 5+2a+2b] \\ \theta(e_5) \leq 25 \\ \theta(e_9) > 20 \end{cases}$$

$$\text{or} \begin{cases} \theta(e_1) \in [10, 20] \\ \theta(e_1) \leq \theta(e_2) \\ \theta(e_2) \in [a, a+b] \\ \theta(e_3) \in [5+a, 5+a+b] \\ \theta(e_4) = +\infty \\ \theta(e_5) \in [5+2a, 5+2a+2b] \\ \theta(e_9) = +\infty \end{cases} \quad \text{or} \quad \begin{cases} \theta(e_1) = +\infty \\ \theta(e_2) \in [a, a+b] \\ \theta(e_2) \leq 20 \\ \theta(e_3) \in [5+a, 5+a+b] \\ \theta(e_4) \in [15, 25] \\ \theta(e_4) \leq \theta(e_5) \\ \theta(e_5) \in [5+2a, 5+2a+2b] \\ \theta(e_9) = +\infty \end{cases}$$

From these constraints, we can for instance deduce that for $e_1$ to occur, we should of course have $a+b \geq 10$ and, for $e_4$ to occur, we should have $a+b \geq 5$ and $a \leq 20$.

Finally, note that, in general, the DOA operator introduces constraints on sums of variables and so does the introduction of parameters. In this example, this simplifies nicely because we have chosen fixed durations for the "freezing" of transition $t_1$, for the sake of the clarity of the presentation.

Given that marking reachability, for instance, is undecidable for safe Sw-PNs [5], we know that a finite complete prefix cannot be obtained but we can still use other analysis techniques like supervision that will constrain the unfolding according to a finite set of observations and indeed give a finite prefix compatible with those observations [17].

## 7 Conclusion

We have proposed a new technique for the unfolding of safe parametric stopwatch Petri nets that allow a symbolic handling of both time and parameters. To the best of our knowledge, this is the first time that the parametric or stopwatch cases are addressed in the context of unfoldings. Moreover, when restricting to the subclass of safe time Petri nets, our technique compares well with the previous approach of [9]. It indeed provides a more compact unfolding, by eliminating the duplication of transitions, and also removes the need for read arcs in the unfolding. As a tradeoff, the constraints associated with the firing times of events may seem slightly more complex.

In the case of TPNs, we have shown how to obtain a finite complete prefix of the symbolic unfolding. This is however not possible in presence of stopwatches or parameters due to known undecidability results on these richer models. It is however possible to couple the method with a supervision technique, based on the idea of [15], that makes the unfolding finite based on a finite set of observations. This approach, that does work with parameters and stopwaches, is detailled in [17] with a case study.

Further work includes a proper implementation (a very preliminary one, without any prefix computation, is available in our tool Roméo [23]), investigating non-safe bounded models and the application of the unfolding technique to revisit the problems of model-checking and control.

## References

1. Abdulla, P.A., Iyer, S.P., Nylen, A.: Unfoldings of unbounded Petri nets. In: Proceedings of CAV, *LNCS*, vol. 1855, pp. 495–507. Springer (2000)
2. Aura, T., Lilius, J.: A causal semantics for time Petri nets. Theoretical Computer Science **243**(2), 409–447 (2000)
3. Baldan, P., Busi, N., Corradini, A., Pinna, G.M.: Functorial concurrent semantics for Petri nets with read and inhibitor arcs. In: CONCUR, *LNCS*, vol. 1877, pp. 442–457. Springer (2000)
4. Berthomieu, B., Diaz, M.: Modeling and verification of time dependent systems using time Petri nets. IEEE transactions on software engineering **17**(3), 259–273 (1991)
5. Berthomieu, B., Lime, D., Roux, O.H., Vernadat, F.: Reachability problems and abstract state spaces for time Petri nets with stopwatches. Journal of Discrete Event Dynamic Systems (jDEDS) **17**(2), 133–158 (2007)
6. Berthomieu, B., Menasche, M.: An enumerative approach for analyzing time Petri nets. In: R.E.A. Mason (ed.) Information Processing: proceedings of the IFIP congress 1983, *IFIP congress series*, vol. 9, pp. 41–46. Elsevier Science Publishers, Amsterdam (1983)

7. Bouyer, P., Haddad, S., Reynier, P.A.: Timed unfoldings for networks of timed automata. In: S. Graf, W. Zhang (eds.) Proceedings of the 4th International Symposium on Automated Technology for Verification and Analysis (ATVA'06), *Lecture Notes in Computer Science*, vol. 4218, pp. 292–306. Springer, Beijing, China (2006)
8. Cassez, F., Chatain, T., Jard, C.: Symbolic Unfoldings for Networks of Timed Automata. In: S. Graf, W. Zhang (eds.) Proceedings of the 4th International Symposium on Automated Technology for Verification and Analysis (ATVA'06), *Lecture Notes in Computer Science*, vol. 4218, pp. 307–321. Springer, Beijing, China (2006)
9. Chatain, T., Jard, C.: Complete finite prefixes of symbolic unfoldings of safe time Petri nets. In: Proceedings of ICATPN, *LNCS*, vol. 4024, pp. 125–145. Springer (2006)
10. Chatain, T., Khomenko, V.: On the well-foundedness of adequate orders used for construction of complete unfolding prefixes. Information Processing Letters **104**(4), 129–136 (2007)
11. Dill, D.: Timing assumptions and verification of finite-state concurrent systems. In: Proc. of Workshop on Computer Aided Verification Methods for Finite State Systems, vol. 407, pp. 197–212 (1989)
12. Esparza, J.: Model checking using net unfoldings. Science of Computer Programming **23**, 151–195 (1994)
13. Esparza, J., Heljanko, K.: Unfoldings, A Partial-Order Approach to Model Checking. Monographs in Theoretical Computer Science. Springer (2008)
14. Esparza, J., Rmer, S., Vogler, W.: An improvement of McMillan's unfolding algorithm. Formal Methods in System Design **20**(3), 285–310 (2002)
15. Fabre, E., Benveniste, A., Haar, S., Jard, C.: Distributed monitoring of concurrent and asynchronous systems. Discrete Event Dynamic Systems **15**(1), 33–84 (2005)
16. Gardey, G., Roux, O.H., Roux, O.F.: State space computation and analysis of time Petri nets. Theory and Practice of Logic Programming (TPLP). Special Issue on Specification Analysis and Verification of Reactive Systems **6**(3), 301–320 (2006)
17. Grabiec, B., Traonouez, L.M., Jard, C., Lime, D., Roux, O.H.: Diagnosis using unfoldings of parametric time Petri nets. In: K. Chatterjee, T.A. Henzinger (eds.) 8th International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS 2010), *Lecture Notes in Computer Science*, vol. 6246, pp. 137–151. Springer, Vienna, Austria (2010)
18. Henzinger, T., Kopke, P., Puri, A., Varaiya, P.: What's decidable about hybrid automata? Journal of Computer and System Sciences **57**, 94–124 (1998)
19. Khomenko, V., Koutny, M.: Branching processes of high-level Petri nets. In: Proceedings of TACAS, *LNCS*, vol. 2619, pp. 458–472. Springer (2003)
20. Lime, D., Roux, O.H.: A translation-based method for the timed analysis of scheduling extended time Petri nets. In: 25th IEEE Real-Time Systems Symposium (RTSS 2004), pp. 187–196. IEEE Computer Society Press, Lisbon, Portugal (2004)
21. Lime, D., Roux, O.H.: Model checking of time Petri nets using the state class timed automaton. Journal of Discrete Event Dynamic Systems (jDEDS) **16**(2), 179–205 (2006)
22. Lime, D., Roux, O.H.: Formal verification of real-time systems with preemptive scheduling. Journal of Real-Time Systems **41**(2), 118–151 (2009)
23. Lime, D., Roux, O.H., Seidner, C., Traonouez, L.M.: Romeo: A parametric model-checker for Petri nets with stopwatches. In: S. Kowalewski, A. Philippou (eds.) 15th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2009), *Lecture Notes in Computer Science*, vol. 5505, pp. 54–57. Springer, York, United Kingdom (2009)
24. Margaria, T., Steffen, B. (eds.): An Improvement of McMillan's Unfolding Algorithm, *Lecture Notes in Computer Science*, vol. 1055. Springer, Passau, Germany (1996)
25. McMillan, K.L.: Using unfolding to avoid the state space explosion problem in the verification of asynchronous circuits. In: Proceedings of Computer Aided Verification (CAV'92), *LNCS*, vol. 663, pp. 164–177. Springer (1992)
26. Merlin, P.M.: A study of the recoverability of computing systems. Ph.D. thesis, Dep. of Information and Computer Science, University of California, Irvine, CA (1974)
27. Traonouez, L.M., Grabiec, B., Jard, C., Lime, D., Roux, O.H.: Symbolic unfolding of parametric stopwatch Petri nets. In: A. Bouajjani, W.N. Chin (eds.) 8th International Symposium on Automated Technology for Verification and Analysis (ATVA 2010), *Lecture Notes in Computer Science*, vol. 6252, pp. 291–305. Springer, Singapore (2010)

28. Traonouez, L.M., Lime, D., Roux, O.H.: Parametric model-checking of stopwatch Petri nets. Journal of Universal Computer Science (J.UCS) **15**(17), 3273–3304 (2009)
29. Vogler, W., Semenov, A.L., Yakovlev, A.: Unfolding and finite prefix for nets with read arcs. In: D. Sangiorgi, R. de Simone (eds.) 9th International Conference on Concurrency Theory (CONCUR'98), *Lecture Notes in Computer Science*, vol. 1466, pp. 501–516. Springer, Nice, France (1998)